RESEARCH IN COMPUTING SCIENCE

ISSN: 1870-4069

Advances in Artificial Intelligence and Applications



Alexander Gelbukh
Ángel F. Kuri Morales
(Eds.)

Vol. 32

RCS
Research in Computing Science

Special Issue: Advances in Artificial Intelligence and Applications

Research in Computing Science

Series Editorial Board

Comité Editorial de la Serie

Editors-in-Chief:

Editores en Jefe

Juan Humberto Sossa Azuela (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Editores Asociados

Jesús Angulo (Frane)
Jihad El-Sana (Israel)
Jesús Figueroa (Mexico)
Alexander Gelbukh (Russia)
Ioannis Kakadiaris (USA)
Serguei Levachkine (Russia)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

Coordinación Editorial

Blanca Miranda Valencia

Formatting:

Formación

Sulema Torres Ramos

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 32**, Noviembre, 2007. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela*, RFC *SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 32**, November, 2007. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, November, 2007, in the IPN Graphic Workshop – Publication Office.*

Volume 32

Volumen 32

Special Issue: Advances in Artificial Intelligence and Applications

Volume Editors:
Editores del Volumen

Alexander Gelbukh
Ángel Fernando Kuri Morales

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2007

ISSN: 1870-4069

Copyright © 2007 Instituto Politécnico Nacional
Copyright © 2007 Instituto Politécnico Nacional

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX and Periódica
Indexada en LATINDEX y Periódica

Printing: 500
Impresiones: 500

Printed in Mexico
Impreso en México

Preface

Artificial Intelligence is a branch of computer science that develops heuristic methods for solving super-complex problems that cannot be solved by direct calculations or mathematics methods, as well as ways to model human intellectual activity. Applications of artificial intelligence provide near-to-optimal solutions of very complex problems in industry and social life, perform large-scale tasks extremely complex for a computer, and help specialists in decision making and knowledge management.

This volume presents 45 original research papers selected after careful review process by the Editorial Board of the volume out of 149 submissions considered, i.e., acceptance rate was 30%.

The volume is structured into seven thematic fields representative of the main current areas of interest within the AI community:

- Computational Intelligence
- Knowledge Representation and Reasoning
- Machine Learning
- Constraint programming, Planning, and Scheduling
- Natural Language Processing
- Intelligent Tutoring Systems
- Intelligent Biomedical Applications

The academic and editorial effort resulting in this volume was carried out in collaboration with, and was supported by, the Mexican Society for Artificial Intelligence (SMIA). We cordially thank all people involved in its preparation. In the first place these are the authors of the papers constituting it: it is the excellence of their research work that gives sense to the work of all other people involved. Our very special thanks go to the members of the Board of Directors of SMIA, especially to José Galaviz Casas. Mikhail Alexandrov, Hiram Calvo, Denis Filatov, Oleksiy Pogrebnyak, Grigory Sidorov, Manuel Vilares were among the most helpful and active members of the Editorial Board. Sulema Torres, Yulia Ledeneva, and Alejandro Peña of the CIC-IPN, Héctor Jiménez Salazar of UAM, and Oscar Celma of the Music Technology Group of the Pompeu Fabra University devoted great effort to the preparation of the volume. We would like to express our sincere gratitude to the IEEE Section of Aguascalientes, the Instituto Tecnológico de Aguascalientes, the Universidad Autónoma de Aguascalientes, the Universidad Tecnológica de Aguascalientes, the Tecnológico de Monterrey Campus Aguascalientes, and the Universidad Politécnica de Aguascalientes for their valuable collaboration. Special thanks to Ing. Luis Armando Reynoso Femat, Dr. José Antonio Calderón Martínez, and all other people involved in the preparation of the volume. The submission, reviewing, and selection process was supported for free by the EasyChair system, www.EasyChair.org.

Alexander Gelbukh
Ángel Kuri

November 2007

Table of Contents
Índice

Page/Pág.

Computational Intelligence

Conditions and Constraints for better Approximate Reasoning with Fuzzy Logic 3
Ruben Ruelas, Donald Carr

Parameter Estimation of BACK equation of Sate Via Gray-Box Neural Network Models 16
Ulises Ivan Bravo Sanchez, Florianne Castillo Borja, Ramiro Rico Martinez, Gustavo Arturo Iglesias Silva

Implementation of Data Fusion in a Probabilistic Neural Network (PNN) for Classification of Fruit Ripening Stages 27
Jorge Isidro Aranda, Arturo Baltazar

Flexural Strength Prediction of Slabs with Finished Type Tile: A Neural Network Approach 38
Jorge Acevedo-Dávila, Luis Torres Treviño

Hardware/Software Partitioning of Coarse-Grained Reconfigurable System Using Evolutionary Ant Colony Optimization 45
Dawei Wang, Yafei Cao, Sikun Li

Discovering Efficient Turbo Code Interleavers by Genetic Algorithms 56
Pavel Krömer, Vaclav Snasel, Nabil Ouddane

A Comparative Study of a Fuzzy Similarity Measure Aggregated with Fuzzy Implications Applied to Shape Recognition 66
Leila Baccour, Wafa Maghrebi, Slim Kanoun, Mohamed Adel Alimi

Uncertainty Management in Context-Aware Applications 77
Pedro Damián-Reyes, Jesús Favela, Juan Contreras-Castillo, JRG Pulido

Knowledge Representation and Reasoning

Basic Transformations that Preserve G'3-stable Equivalente 89
José Luis Carballido, Mauricio Osorio, José Arrazola

Rewards in Persuasive Group Decision Making 99
Ricardo Santos, Goreti Marreiros, Carlos Ramos, José Neves, José Bulas-Cruz

Representation and Reuse of Design Knowledge and Design Rationale for Design Decision-Making 111
Jie Hu, Yinghong Peng, Guangleng Xiong

Modeling Ontology-Based Task Knowledge for PMBOK: A Three-Layer Method 120
Kwoting Fang, Chingwei Chang, Yenping Chi

Analysis and Classification of Car Engine Sounds to Diagnose Failures 131
Luis Andrés Méndez-Ortega, Orion Fausto Reyes-Galaviz

Study on the Multiagent Based Model System and Model Integration 141
Daiping Hu, Ruiming Wu, Mengyu Li, Qi Wang
Defeasible Decision Making in a Multi-Robot Environment 150
Edgardo Ferretti, Nicolas Rotstein, Marcelo Errecalde, Alejandro Garcia, Guillermo R. Simari
Normative Conflict Detection for e-Contracts 161
Georgios K. Giannikis, Aspassia Daskalopulu

Machine Learning

Tri-Testing: A Novel Semi-Supervised Learning Method based on Ensemble Learning and Active Learning 175
Shota Kubo, Masahiro Terabe, Kazuo Hashimoto, Ryuta Ono
Statistical Selection of Relevant Features to Classify Random, Scale Free and Exponential Networks 185
Tania Turrubiates López, Claudia Gómez Santillán, Laura Cruz Reyes, Rogelio Ortega Izaguirre, Eustorgio Meza Conde
A First Approach to Understanding the Anomalies on My PC 193
Luis M. Fernandez, Hugo Terashima-Marín, Manuel Valenzuela-Rendón
Modelling and Prediction of the MXNUSD Exchange Rate Using Hybrid RLS-BP Interval Type-1 Non-Singleton Type-2 Fuzzy Logic Systems 204
Gerardo M Mendez, Angeles Hernandez
Entropy Based Linear Approximation Algorithm for Time Series Discretization .. 214
Acosta-Mesa Héctor-Gabrie, Cruz-Ramírez Nicandro, García-López Daniel-Alejandro

Constraint Programming, Planning, and Scheduling

Process Calculi for Adaptive Enumeration Strategies in Constraint Programming 227
Eric Monfroy, Carlos Olarte, Camilo Rueda
Variable and Value Selection Heuristics: Application to Solve Puzzles 238
Broderick Crawford, Mary Aranda
Comparison between Genetic Algorithms with Exaptation and Case Injected Genetic Algorithm in dynamic job shop problems 249
Luis Torres Treviño
Research on Dynamic OD Analysis for Urban Traffic 258
DONG Hong-zhao, ZHOU Min, GUO Ming-fei, Qian Xiao-hong
Case-Based Route Planning in the Mexico City Subway System 269
Andrés Gómez de Silva Garza, Lourdes Domenzain Ortega
Analysis of the Simulated Annealing Convergence in Function of the Standard Deviation and the Boltzmann Quotient for Scheduling Problems 282
Martín G. Martínez-Rangel, Marco Antonio Cruz-Chávez, José Crispín Zavala-Díaz, David Juárez-Romero, Ocotlán Díaz-Parra

Natural Language Processing

Comparison of Three Algorithms for Automatic Keyword Extraction 297
Mayra A. Paredes-Farrera, Hiram Calvo
Homonymy Resolution and Tagging of Grammar Classes without Lexical Information using Neural Networks* 305
Ernesto Gómez-Balderas, Grigori Sidorov, Héctor Jiménez-Salazar
Extract Conceptual Graphs from Plain Texts in Patent Claims 315
Shih-Yao Yang, Von-Wun Soo
A Construction of Emotion Thesaurus Basing on Chinese Character and Empirical Knowledge 330
Yu Zhang, Zhuoming Li, Fuji Ren, Shingo Kuroiwa
Music Composition using the EMILE Grammar Inductor 341
David Ortega-Pacheco, Hiram Calvo
Managing the Workflow of Massive Feeding of Digital Libraries 352
Angeles S. Places, Nieves R. Brisaboa, Jose R. Parama, Oscar Pedreira, Diego Seco
English Speech to Sanskrit Speech Translator 363
Abhradeep Guha Thakurta, Ashwath Kumar K., Hashir Khan
Color Features Based Speaking Detection with Hidden Markov Model in Video Sequences 374
Peilin Jiang, Ran Li, Fuji Ren, Shingo Kuroiwa, Nanning Zheng

Intelligent Tutoring Systems

Authoring Fuzzy-based Tutoring Systems for M-Learning 385
Ramón Zatarain, M. L. Barrón-Estrada, Guillermo A. Sandoval-Sánchez, L. Vega-Zazueta, Carlos A. Reyes-García
An Approach to Learning Objects Classification using an Auto-Associative Neural Network 395
Clara Pérez-Molina, Miguel Rodríguez-Artacho
L2Code: An Author Environment to Integrate Multiple Intelligences and Naive Bayesian Classifiers for MultiParadigm-Programming Training 406
M. L. Barron-Estrada, Ramon Zatarain-Cabada, J. Moises Osorio-Velazquez, L. Zepeda-Sanchez, Carlos A. Reyes-Garcia

Intelligent Biomedical Applications

Pre Diagnostician of Heart Arrhythmias Using Wavelet Transform and Fuzzy c-Means 419
Ramón Miranda, Isaac Chairez, Agustín Cabrera
Hierarchal Fuzzy Logic Medical Database with Decision Algorithm for Metabolic Syndrome Diagnosis 430
I. Salgado, A. Rodríguez, J. Chairez, A. Zúñiga, P. Santillan
Rule-Based Expert System to Determine the Likely Diseases in an Air-Polluted Environment 442

R. V. Díaz, Zenaida García, Mateo Lezcano, Elena Rosa Dominguez
Using Formal Concept Analysis to Leverage Ontology-Based Foot Acu-Point Massage Knowledge System 450
Kwoting Fang, Chingwei Chang, Yenping Chi
Descriptive Characteristics Generation and Selection for Acute Leukemia Subtype Classification from Bone Marrow Digital Images 460
M. Coral Galindo Domínguez, Jesús A. González, Leopoldo Altamirano Robles, Ivan Olmos Pineda
An Electronic Game with Speech Recognition for Children with Problems of Speech 471
Carlos Miranda-Palma, Reyna Camal-Uc, José Cen-Magaña, Cinhtia Gonzalez-Segura, Sergio Gonzalez-Segura, Manuel Escalante, Jorge Ríos
Simulating Lumbar Spine Motion: Parameter Estimation for Realistic Modeling 480
Ahmed Sameh

Author Index 491
Índice de autores

Editorial Board of the Volume 493
Comité editorial del volumen

Additional Reviewers 493
Árbitros adicionales

Computational Intelligence

Conditions and Constraints for better Approximate Reasoning with Fuzzy Logic

Rubén Ruelas, Donald W. Carr Finch

DIP-CUCEI Universidad de Guadalajara
José Guadalupe Zuno No. 48. C.P. 45101 Zapopan, Jalisco, México.
rruelas@newton.dip.udg.mx, doncarr@gmail.com

Abstract. Fuzzy sets and fuzzy logic were proposed as a means to handle states and/or relations that are not well defined or that present gradual transitions. These characteristics have enabled simpler solutions to numerous problems, particularly in the area of control. Nevertheless, the model used for most applications does not fully match the requirements for fuzzy logic. Though the model is fuzzy, because it is based on fuzzy sets and fuzzy operators, it is not a logic model. Also, fuzzy models are generally more important from a numeric approximation point of view. In this paper we give some conditions and constraints for better approximate reasoning with a fuzzy logic model, which is more important from a linguistic approximation point of view. However, the monotonic property is necessary, otherwise, when this condition is not fulfilled, it can be used as exceptions or as approximate reasoning with decision making.

Keywords: Fuzzy Logic, Approximate Reasoning, Fuzzy Rules, Human-Machine Interface.

1 Introduction

Since fuzzy sets and fuzzy logic were proposed, they have been used as an excellent tool to handle states and/or relations that are not well defined or have gradual transitions among other characteristics [1, 2]. These characteristics enabled simpler solutions to numerous problems, particularly in the area of control [3, 4]. Nevertheless, for most applications the model does not fully match the requirements of fuzzy logic. Therefore, the model is fuzzy, because it is based on fuzzy sets and fuzzy operators, but it is not logic because the inference with these models differs from the inference in logic. Besides this, these models are generally more important from a numeric approximation point of view [5].

A strictly fuzzy logic model has little application even if some efforts have been made to use it as it has been used for fuzzy models in control. The main problem arises at the moment of approaching either the conclusion of the fuzzy relation representing the fuzzy rules, or approaching the conclusions which result from the application of the fuzzy rules individually. In both of these cases, it generally results in a subnormal conclusion with a significant degree of uncertainty for a complete rule set, resulting in a conclusion that generally has little meaning. Obviously, because of this, it is difficult to establish a complete set of fuzzy rules that satisfy all the conditions

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 3–15

Received 27/07/07
Accepted 31/08/07
Final version 21/09/07

required for inference, and that allow obtaining the maximum significance for the conclusions of the fuzzy logic model.

This problem of the fuzzy logic model can be explained in the following way. When two fuzzy rules are fired at the same time, but these rules contain contradicting conclusions, the result could be an empty conclusion. However, the necessary conditions required for the parameters of the fuzzy logic model, as well as the restrictions on the maximum uncertainty on the input values to the model, can be very well established. This model can be used for approximate reasoning, and, as more knowledge is involved (more rules are fired at the same time), the conclusion is more precise [6, 7].

In this paper a fuzzy logic model is considered. We show that with this model the most important aspect is the inference using linguistic terms, which serve as labels of the fuzzy sets. Under these conditions, and if the model is well designed, the gradual transition in the conditions of the rules helps to obtain a gradual transition on the terms involved in the conclusions. This avoids the uncertainty other than the fuzziness. This is a natural situation because the uncertainty in the fuzzy rules and in the input data have been taken into account in advance, so better conclusions result from complete input data and from the maximum amount of knowledge (rules) involved in their calculations.

As is shown in this work, the fuzzy sets of the input variables must be defined according to the input data. If they consist on precise values only, the fuzzy sets can be defined as a strict possibilistic partition. Otherwise, for fuzzy input data, the partition must take into account the maximum fuzziness or imprecision in order to define the fuzzy partition of the inputs. We also consider the case of incomplete input and we propose a non standard procedure for the calculation of the output.

One necessary condition, in order to obtain the previous results, is that the model must be structured for monotonic knowledge. When this condition can not be satisfied, because there are rules with similar conditions but very different conclusions, we propose the use of rules that do not satisfy the monotonic property as exceptions. For this, we handle them in a different way, keeping monotonic knowledge in one fuzzy logic model and exceptions in another model, such that the conclusion of the rules are: *y is B unless y is* βC, where *y is* βC represents an exception and β it the degree of certainty. So, the goal of this work is to present the conditions and constraints in order to use the fuzzy logic model for better approximate reasoning.

Section two presents the fuzzy logic model, and in section three we use this model to do approximate reasoning. As this model has monotonic rules, we consider the linear and nonlinear cases as well as the precise and fuzzy inputs, and the case of incomplete input. In this section we also handle as exceptions the rules that do not satisfy the monotonic conditions and we propose a more general output for a fuzzy logic model. Finally, in section four, we give the main conclusions concerning this work.

2 Fuzzy Logic Model

Fuzzy logic was proposed as a means to do reasoning under uncertainty. This can be done with a fuzzy model based on a set of conditional fuzzy rules such as "*if ... then ...*". The quantification of the condition is made with the observation *x is A'* and the

inference follows the Generalized Modus Ponens (GMP) [2], such that a conclusion *y is B_j'* results for each rule, as shown in the next scheme.

Rule:	If x is A_i	Then y is B_j
Observation:	x is A'	
Conclusion:		y is B'

A rule is modeled by a relation that depends on an implication operator. In the case of fuzzy logic there exist several possibilities and the choice can be made by the properties and/or the behavior of the operator [8]. In fact, this has resulted in a typology of fuzzy rules [9]. In this work we consider certainty rules, which express that y is in B_i only for input values x in the core of A_i.

From the computational point of view, a fuzzy rule is modeled by a fuzzy relation $R_k = A_i \rightarrow B_j$, where A_i and B_j are the fuzzy sets of the input and output variables respectively. The calculation of a conclusion B' using this relation is $B' = A' \circ R_k$, that is, it is calculated according to $\mu_{B'}(y)=sup_{x \in X} T[\mu_{A'}(x), \mu_R(x,y)]$, $\forall y \in Y$, where the relation $\mu_R(x,y)$ represents the conditional rule, and T a T-norm. Fig. 1 shows an example of fuzzy rule based on the Kleene Dienes implication $(\mu_R(x,y)=min(1-\mu_A(x), \mu_B(y)))$.

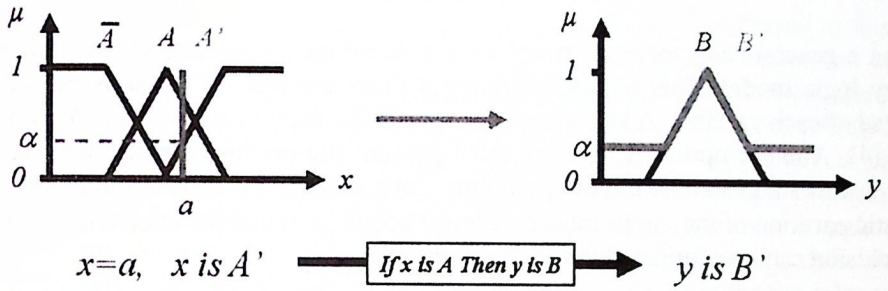


Fig. 1. Inference using a fuzzy rule based on the Kleene Dienes implication.

According to the MPG, as the observation *x is A'* approaches the premise *x is A*, and for certainty rules it approaches the core of *A*, the conclusion approaches *B* with a reducing uncertainty level υ. For the Kleene Dienes operator, the uncertainty υ corresponds to $\upsilon = sup_{x \in X} min(A'(x), 1-A(x))$ [10]. From this expression it is clear that this value will be zero iff the complement of the fuzzy set $A(x)$ is defined outside of the support of $A'(x)$, or $A'(x) \subseteq core\ of\ A(x)$.

From the logical point of view, given A_k the property $B_k = \cap_{i=1,n} A_k \circ (A_i \rightarrow B_i)$ must hold for a set of implicative fuzzy rules. And, if A_k is defined such as $A_k \subseteq A_i \cap A_{i+1}$, the conclusion must be a logical output satisfying the previous property where $B_k \subseteq B_j \cap B_{j+1}$, according to the Generalized Modus Ponens.

In other words, take A_k such that the membership degree to A_i is equal to one, and the membership degree to A_{i+1} is equal to α, with $\alpha \in (0,1)$, then the conclusion is in this case $B_k = \cap_{i=1,n} A_k \circ (A_i \rightarrow B_j)$, with $B_k \subset B_j$. The degree of approximation between these fuzzy sets depends on the α value. For example, as $\alpha \rightarrow 0$, $B_k \rightarrow B_j$, whereas, as $\alpha \rightarrow 1$, $B_k \rightarrow B_j \cap B_{j+1}$.

As can be seen in Fig. 1, the conclusion B' is a normal fuzzy set similar to B, and it presents the uncertainty υ over all the universe of discourse. The shape of B' and its similarity to B change according to the similarity between A and A' and the implication operator used. Also, as the final conclusion must be determined by the conjunction of the individual conclusions, this conclusion will be normalized only if the fuzzy partition of the output was built in order to obtain this result [7].

When using a fuzzy logic model we can follow two approaches. One of the approaches is to combine in a unique fuzzy relation the relations given by the individual fuzzy rules as given in (1) and (2). The other approach is to combine the results given by inference with individual fuzzy rules, as given by (3). The main differences between both approaches are the operating mode and the final conclusion. For the former model, the conclusion is more precise [11]. However, in this work we limit the analysis to models with individual rules.

$$R = \cap_{k=1,n} R_k \tag{1}$$

$$B' = A' \circ R \tag{2}$$

$$B' = \cap_{k=1,n} A' \circ R_k \tag{3}$$

As a general case, there is always interest to define a complete set of rules for a fuzzy logic model. This requires defining a fuzzy partition of the universe of discourse of each variable. A strict fuzzy partition is the most common case, and it satisfies (4). Another option is to use a strict possibilistic partition, which satisfies (5). Fig. 2 shows an example of both partitions. For a fuzzy logic system, the strict possibilistic partition of the inputs must be a lower bound such that the uncertainty υ of the conclusion can be eliminated for coherent reasoning.

$$\sum_{i=1}^{n} \mu_{A_i}(x) = 1, \quad \forall x \in X \tag{4}$$

$$\bigvee_{i=1}^{n} \mu_{A_i}(x) = 1 \quad and \quad \sum_{i=1}^{n} \mu_{A_i}(x) < 2, \quad \forall x \in X \tag{5}$$

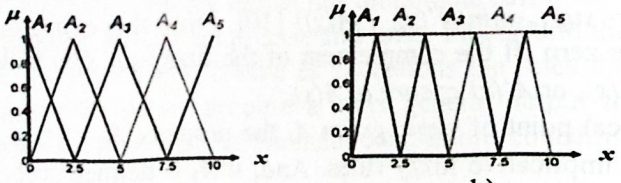


Fig. 2. a) Strict fuzzy partition, b) Strict possibilistic partition.

According to the previous analysis of the model, the fuzzy partition of the inputs must be made such that they allow better conclusions and a minimum of uncertainty in the result. Comparatively, the intersection of two adjacent fuzzy sets must be placed at the invariants, or at the local minima, in a similar way as a fuzzy model built

for function approximation [12]. On the other hand, the fuzzy sets of the output must have at least the minimum intersection such that every conclusion given by the model is normalized. In other words, if p is the maximum number of different conclusions given by rules that could be activated at the same time, this is the minimum number of output fuzzy sets that must have a non empty intersection in order to satisfy this condition.

3 Approximate Reasoning

As it was shown in [7], the fuzzy model must have possibilistic partitions in the input universe of discourses, but the output needs a partition that satisfies the minimal number of intersection fuzzy sets. In this work we consider more restrictions and conditions in order to take better advantage of the fuzzy logic model. These results improve some results presented in [13, 14], and generalize to the multi-input model. For instance, take the next rule as an example of fuzzy rule

$$If\ x_1\ is\ A_i\ and\ x_2\ is\ B_j\ Then\ y\ is\ C_k \tag{6}$$

where $i = 1, 2, ..., N_1, j = 1, 2, ..., N_2$, and $k = i+j-1$. The rules expressed in this way refer to monotonic knowledge so the relation among inputs and output can be expressed as a table of rules, as shown in Table 1. In order to simplify the analysis of the model we consider a monotonic reasoning. When this condition is not satisfied, the non-monotonic part is taken into account in a parallel model with a model having monotonic knowledge and another model with the rules that do not satisfy monotonic condition. For example, take the previous rule which supposes the order $C_{k-1} < C_k < C_{k+1}$ on the fuzzy sets of the output. Under this consideration, the definition of the fuzzy sets is simplified because we have the same conclusion for the rules where the sum $(i+j)$ is constant. See Table 1.

Table 1. Monotonic rules for the fuzzy logic model.

$X_2 \backslash X_1$	BN	SN	ZE	SP	BP
BP	C_5	C_6	C_7	C_8	C_9
SP	C_4	C_5	C_6	C_7	C_8
ZE	C_3	C_4	C_5	C_6	C_7
SN	C_2	C_3	C_4	C_5	C_6
BN	C_1	C_2	C_3	C_4	C_5

The monotonic condition, accompanied with the design of the fuzzy sets for the model, is a necessary and sufficient condition in order to have coherent reasoning. If this condition can not be satisfied, it is necessary to divide the model into sub-models and to build a meta-model such that it will be the responsible for determining the final output, according to the output provided by each model, or if it is possible to consider the non monotonic behavior as exceptions as proposed below.

Taking into account the previous considerations, the necessary fuzzy sets for the output can be calculated as $n_z=(1-N)+\Sigma_{j\ 1..N} n_j$, for N inputs, where n_z and $n_j, j=1,2, \ldots, N$, are the number of fuzzy sets for the output and the inputs respectively. So, let us consider a model with two inputs having strict possibilistic partitions. Every input value has no more than two non-zero membership degrees in accordance with fuzzy sets. Take i and $i+1$ for input one, and j and $j+1$ for input two for example. Then, the maximum number of rules activated at the same time is four, but their conclusions use only three fuzzy sets, that is $k_1=i+j-1$, $k_2=i+j$ and $k_3=i+j+1$.

An example of fuzzy sets for this model is given in Fig. 3. When a strict possibilistic partition is used in the inputs, the model is designed in order to work only with precise values. In this case, in the partition of the output there must be the intersection of at least three adjacent fuzzy sets. This implies that the maximum number of rules fired at the same time is four but the number of different conclusions is three.

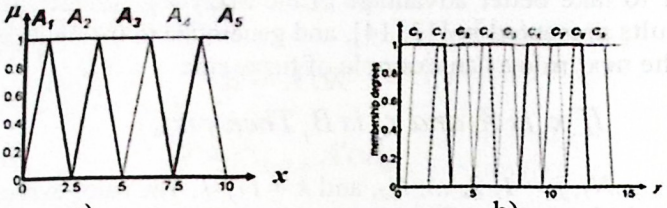


Fig. 3. a) Strict possibilistic partition for the inputs. b) Possibilistic partition for the output.

Linear Reasoning

According to the condition for the fuzzy sets of the output, and as the model is more appropriate for an approximate reasoning based on symbols, the partition of the output must be defined based on the intersection of the fuzzy sets. This is shown in Fig. 4a, where two hypothetical fuzzy sets (C_0 and C_{10}) are considered at the ends in order to define completely the fuzzy sets. They provide the internal limit of the C_2 and C_8 fuzzy sets.

Based on these fuzzy partitions, the fuzzy logic model gives approximate reasoning for three kinds of symbols as those illustrated in Fig. 4b. This also shows that the most precise linguistic conclusions of the model are the fuzzy sets $C_i \cap C_{i+1} \cap C_{i+2}$. Take for example the conclusions of two individual rules such as C_3 and C_4. Their aggregation is a normal fuzzy set. If the conclusion is given by four fuzzy rules, say C_1, C_2, C_2 and C_3 for example, the result continues to be a normal fuzzy set. If only one rule is fired and the conclusion is C_7 for instance, the result is normal and more imprecise. The main difference between these examples is that when more knowledge is taken into account, more rules are fired, and the more precise the final conclusion becomes.

Linear reasoning is considered a fuzzy logic model with uniform strict possibilistic partition of the inputs, and uniform possibilistic partition based on the most precise fuzzy sets of the output, as those of the Fig. 4a. Fig. 5 shows an example. So, as more rules are involved in the final conclusion, they serve as a factor to increase, to some degree, the precision of the final conclusion. The increase in precision depends on the certainty of the activated rules.

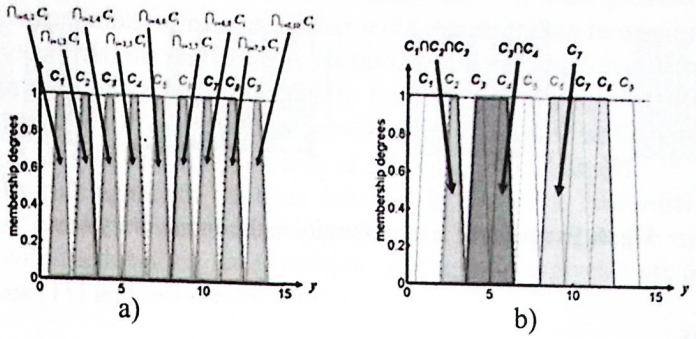


Fig. 4. a) Most precise fuzzy sets that defines the partition of the output. B) Different fuzzy sets that can be given as result of reasoning.

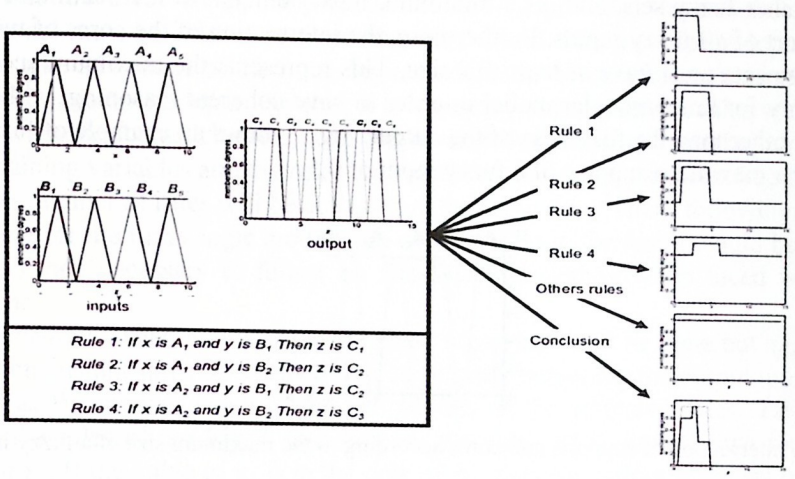


Fig. 5. Fuzzy logic model with linear reasoning.

Nonlinear Reasoning

In the previous section we consider monotonic and linear reasoning. However, fuzzy logic is interesting because nonlinearity can be handled in a natural way. This can be observed in the perception, reasoning and decision making of human beings, and this is what fuzzy logic was developed to model [15]. So, if fuzzy rules are defined according to the monotonic property, the most difficult task is to find the appropriate distribution of the fuzzy sets. The fuzzy sets of the inputs and the output must take into account the nonlinearity of the reasoning process in the space where the model is defined. Fig. 6 contains some examples for the output fuzzy sets $C_i \cap C_{i+1} \cap C_{i+2}$. In Fig. 6 there are some regions with more precise and some regions with more imprecise information, the former represented by narrower fuzzy sets. Obviously, the nonlinearity of reasoning can also be done by the non uniformity of the partition of the inputs. Nevertheless, they are not shown here for simplicity.

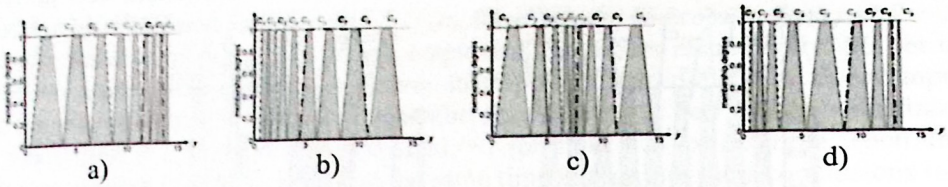


Fig. 6. Examples of non uniformity in the output fuzzy sets.

Fuzzy Inputs

The partition on the inputs depends on the application. For precise inputs the best is a strict possibilistic partition, whereas for fuzzy inputs the partition must not be strict. Although the core of each fuzzy set must have a minimum intersection with the core of other fuzzy sets, and, as a minimum, must be equal to the maximum size of the support of all fuzzy inputs. Furthermore, the intersection of the cores of two adjacent fuzzy sets must have at least this size. This represents the maximum acceptable uncertainty for this particular model in order to have coherent reasoning, without uncertainty other than the fuzziness of the output. Fig. 7 shows an example of fuzzy partition for a maximum support of a fuzzy input of 0.5.

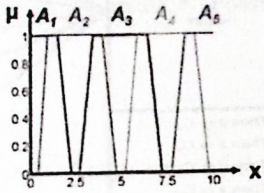


Fig. 7. Intersection of supports and cores according to the maximum size of a fuzzy input.

Consider the maximum imprecision allowed at the inputs. As shown in the Fig. 7, the minimum separation between the supports of two non-adjacent fuzzy sets of the inputs must then be this maximum imprecision. Thus, for every input value at most two fuzzy sets have a non-empty intersection. Also, the intersection of the cores of two adjacent fuzzy sets must be equal or greater than the maximum imprecision.

Taking into account the uncertainty (fuzziness or imprecision) of the input, the fuzzy sets of the inputs can be defined according to this uncertainty. However if the input is within the set of elements of only one fuzzy set, or is within the core ($A_i \cap A_j$), there will be no distinction among precise, imprecise and fuzzy values of the input. This can be interpreted as a degree of distinction which has a strict possibilistic partition for one side (total distinction), and a qualitative partition for the other side (zero distinction). Besides, as the input takes on a fuzzier or a more imprecise value, the model generally gives a more imprecise output.

Designing the partition of the inputs for a maximum fuzziness of the data, each of the inputs will activate a maximum of two rules. Otherwise, if we allowed the input to activate three fuzzy rules or more per input, the conclusions will be much more imprecise and it will require the intersection of four or more fuzzy sets. Thus, this will

require bigger supports and cores for these fuzzy sets. So, if the previous condition is not satisfied, the number of rules activated will be greater than two and it is necessary to make some restrictions on the fuzzy partition of the output. Nevertheless, it is more difficult to satisfy this requirement than to limit the uncertainty of the inputs. Under these conditions $A_i \cup A_j$ can not be accepted as input, because in this case the conclusion of the model will be Y (the universe of discourse of the output).

The rule-by-rule approach has been followed in this work. Nevertheless, if the inputs can have fuzzier values than anticipated, then first aggregate the rules and afterwards infer with the resulting fuzzy relation. This approach gives more precise results for these cases [11] and this can be helpful.

Incomplete Input

As previously shown, for complete input, the conclusions become more precise, as more rules are involved in the particular reasoning process. Nevertheless, a fuzzy logic model must be able to do reasoning when faced with incomplete input. A proposal to take advantage of the fuzzy logic model is to do incomplete reasoning when faced with incomplete input. For this, it is necessary to consider the rules while ignoring the variable for which there is no input value available, and to take into account the remaining variables and the rules that contain them in the premises. So, given this, a greater number of rules will be activated at the same time. Then, following the same procedure for the fuzzy logic model with complete input, the result would be incoherent. It is then necessary to follow an alternative procedure when faced with these situations.

For example, when inputs are incomplete, reasoning must be done but aggregation of the conclusions must be avoided. For incomplete reasoning the output must be calculated as the median value of the conclusions of the activated rules. Take for instance, the rules of the Fig. 8 and suppose the value of x_2 is unknown for a given value of x_1. If the value of x_1 is in the core of A_1, then the output is the set of conclusions C_1 to C_5. The median value, in this case, is C_3. If the value of x_1 is in the core of A_2, then the conclusions is the set C_2 to C_6. The corresponding median value is C_4. However, if the input value is in the core of A_1 and A_2, then the set C_1 to C_6 is given as the conclusion. The median value in this case is *C_3 and C_4* which can be translated into a mathematical expression as $C_3 \cap C_4$. This can be the basis for an iterative process of looking for a good conclusion through an approximate reasoning process, waiting for the missing input values. This could also be helpful for the application of the fuzzy logic models in research engines.

As can be seen, the monotonic behavior of the model is not lost when faced with incomplete input. What are lost are the most precise symbols of the output, because in the previous example the conclusion could be two kinds of fuzzy sets but not the most precise ones, as shown in Fig. 4b. So, for these conditions the output can be calculated as the median value that corresponds to a fuzzy set of the output for an odd number of conclusions, and an intersection of two fuzzy sets for an even number of conclusions.

However, the model can still be useful in situations where data is incomplete. For example, for a two input and one output model, if we have an input value for one variable but the value for the other variable is not available at the moment of the in-

ference, it seems interesting to obtain an output (to take advantage of the knowledge coded in the fuzzy model), and to obtain an output that is a suggestion, not a conclusion. Consider a model with three inputs,

$$If\ x_1\ is\ A_i\ and\ x_2\ is\ B_j\ and\ x_3\ is\ C_k\ Then\ y\ is\ D_m \tag{7}$$

where $m = i+j+k-2$. If only x_1 is available, the conclusion is $y = \{D_i, D_{i+1}, \ldots D_{i+N2+N3-2}\}$. The final result can be approached following the previous procedure. When two inputs are available, the conclusion is $y = \{D_{i+j-1}, D_{i+j}, \ldots D_{i+j+N3-2}\}$. Finally, consider the values when all three inputs are available. In this case, the conclusion is $y = D_{i+j+k-2}$. This example clearly shows that when the input information is more complete, the conclusion is more precise.

Non monotonic Reasoning

Monotonic reasoning, linear or nonlinear, presents a limitation as it does not offer much flexibility in the definition of the fuzzy rules, as shown in Fig. 8. In this situation the conclusions of adjacent rules must have a normal intersection, so the implemented reasoning is of the type "More (Less) ... More (Less) ..." or any combination of these. Under this point of view, the fuzzy logic model seems to be a detailed description of a gradual rule "More ... More ...". In order to relax the condition of adjacent conclusions for adjacent rules, the fuzzy sets of the output are harder to define and it becomes more complex as further fuzzy sets must have a non-empty intersection.

Considering regions that go beyond the region where a fuzzy logic model is defined, we can find each region with monotonic knowledge but with non monotonic knowledge among regions. A multi-model approach can be applied such that the behavior is well represented by fuzzy logic models. However, if a domain presents a monotonic behavior, except for particular cases that do not satisfy this condition, the model can be extended to a model with exceptions and the conclusions of the rules modified such that *"y is B unless y is βC"*, where *"y is βC"* represents the exception to the monotonic condition, and β the certainty degree of the exception.

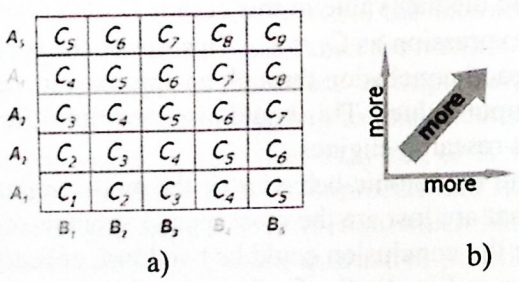


Fig. 8. a) Rules for monotone reasoning b) Gradual reasoning more (less) ... more (less) ..

The fuzzy rules of a model, including an exception, are represented in Fig. 9a, although this model can be decomposed into two fuzzy logic models in parallel. One of the models with a complete set of monotonic rules, as those of the Fig. 8, and the

other with the exception as in Fig. 9b. The final output of the model will be the conclusion of the rules or the exception, depending on the value of the degree of certainty.

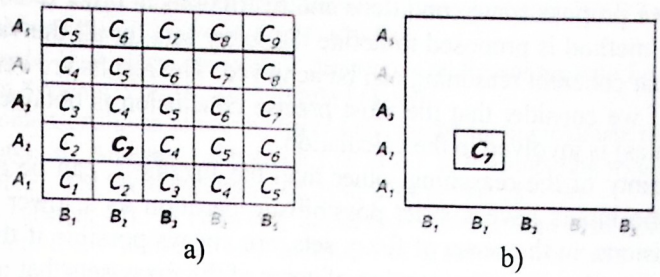


Fig. 9. Non monotony in a rule set.

Calculus of Linguistic and Numeric Output

The output of a fuzzy logic model approximates fuzzy sets with different precision; Fig. 4b shows some examples. However, depending on the application of the fuzzy logic model, it is necessary to approximate the output by the linguistic terms of the possible output; C_i, $C_i \cap C_{i+1}$, $C_{i-1} \cap C_i$, $C_{i-1} \cap C_i \cap C_{i+1}$, etc., and a set of qualifying terms, such as almost, approximately, more or less, etc., that grades the precision of the linguistic term with the conclusion of the model, such that the conclusion would be *almost* C_i, *almost* $C_{i-1} \cap C_i$, *approximately* $C_{i-1} \cap C_i$, *approximately* $C_{i-1} \cap C_i \cap C_{i+1}$, *more or less* $C_i \cap C_{i+1}$, etc. It could also be necessary to approximate the output by numeric values of the universe of discourse Y. A fuzzy logic model can provide both types of values although the linguistic output is the most interesting from the logic point of view.

The linguistic approximation can be used with other models or as a human-machine interface, whereas the numerical approximation can be used with models developed to control systems for example. The fuzzy logic model is then very useful for logical reasoning and numerical approximation, so the model offers the possibility of a good relationship between meaning and precision.

The numerical approach consists in the calculation of a precise value from the fuzzy conclusion of the model using a center of area defuzzification method for example [16]. Consider the fuzzy conclusion shown in Fig. 5. In this case, a defuzzification method, as the one mentioned previously, can be applied to calculate a numerical output for the model. From the design of this model it is possible to get a good numerical approximation, which shows that even if the fuzzy logic model is more appropriate for linguistic reasoning, it is able to easily interface with models where precision is required.

4 Conclusions

The main problem with fuzzy logic models is in the ability to obtain coherent outputs. In this paper we propose some conditions and restrictions in order to solve this problem. Indeed, a method is proposed to define the fuzzy sets for all the variables of the model, such that coherent reasoning can be achieved. The results are very interesting, and, overall, if we consider that the most precise conclusion is obtained when more knowledge (rules) is involved in the calculation.

The uncertainty of the reasoning, other than the fuzziness, can be avoided if the partitions of the inputs have a strict possibilistic partition as a lower bound. Also, normal conclusions, in the sense of fuzzy sets, are always possible if the partition of the output respects the minimum number of cores of the fuzzy sets that must intersect. Linear and nonlinear reasoning can be made with the fuzzy logic model according to the distribution of the fuzzy sets in their respective universe of discourse; however, obviously, linear reasoning gives the simplest partition and distribution.

Knowing the maximum uncertainty of fuzzy inputs it is possible to determine the parameters for the model, in particular, the fuzzy partition of the inputs, in order to obtain coherent reasoning for these kinds of inputs. Also, the knowledge coded in the model can be used when faced with incomplete input. Nevertheless, the model requires of a different procedure for the calculation of the output, and the result is considered a suggestion and not a conclusion.

Non-monotonic behavior is difficult to model with a fuzzy logic model, particularly for the aggregation of the conclusions of the rules. In this work it was proposed that the monotonic behavior should be kept in one model and the cases of non-monotonic behavior be kept in another model such that the final conclusion can be modified to *y is B unless y is* βC, which takes into account the monotonic behavior subject to the exception.

The fuzzy logic model is well adapted in order to make a good connection between models in different levels of a hierarchy, because approximate reasoning can be made from the linguistic or numeric point of view. Linguistic reasoning requires the approximation of the conclusion with the linguistic terms of the fuzzy sets, and a set of qualifying terms such as almost, approximately, more or less, etc. Linguistic reasoning can be used for a higher level of reasoning in an intelligent system, where meaning is more important than precision; this could be the case of a human-machine interface for example. The numeric reasoning requires a defuzzification method, as the center of area method, which could be applied to the conclusions so that precise values can be given and used as input to another fuzzy model. So, the fuzzy logic model described here provides a good relationship between meaning and precision.

References

1. Zadeh, L. A.: Fuzzy sets. Inform. Control, Vol. 8, (1965) 338-353
2. Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on Systems, Man and Cybernetics, Vol. 3, (1973) 28-44

3. Mamdani, E.H., Asilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. Int. J. Man-Mach. Studies, Vol. 7 (1975) 1-13
4. Sugeno, M., Nishida, M.: Fuzzy Control of Model Car. Fuzzy Sets and Systems, Vol. 16 (1985) 103-113
5. Castro, J. L.: Fuzzy Logic Controllers Are Universal Approximators. IEEE Trans. On Syst., Man, and Cybernetics, Vol. 25, No. 4, (1995) 629-635
6. Dubois, D., Prade, H., Ughetto, L.: A New Perspective on Reasoning with Fuzzy Rules. Int. J. of Intelligent Systems, Vol. 18, No. 5 (2003) 541-567
7. Ruelas, R.: Some Conditions to use the fuzzy logic model for approximate reasoning. Jamshidi, M., Reuter, M., Andina, D., Jamshidi, J. S. (eds.): Soft Computing with Industrial Applications, Vol. 17. TSI Press, Albuquerque, NM, USA (2004) 585-590
8. Demirli, K., Turksen, I.B.: A review of implication and the Generalized Modus Ponens. Proc. of the Third IEEE Conf. on Fuzzy Systems, Vol. 2, (1994) 1440-1445
9. Dubois, D., Prade, H.: What Are Fuzzy Rules and How to Use Them. Fuzzy Sets and Systems, Vol.84, (1996) 169-185
10. Kerre, E. E.: A comparative study of the behavior of some popular fuzzy implication operators on the generalized modus ponens. Zadeh, L. A., Kacprzyk, J. (eds.): Fuzzy Logic for the Management of Uncertainty. John Wiley & Sons (1992) 281-295
11. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions. Fuzzy Sets and Systems, Vol. 40, (1991) 143-202
12. Kosko, B.: Optimal fuzzy rules cover extrema. Int. J. of Intelligent Systems, Vol. 10 (1995) 249-255
13. Dubois, D., Prade, H.: Fuzzy Rules in Knowledge Based Systems. Modeling Gradedness, Uncertainty and Preference. Yager, R.R., Zadeh, L.A. (eds.): An Introduction to Fuzzy Logic Applications in Intelligent Systems (1992) 45-68
14. Dubois, D., Prade, H., Ughetto, L.: Coherence of Fuzzy Knowledge Bases. Proc. of the Fifth IEEE Int. Conf. on Fuzzy Systems, Vol. 3 (1996) 1858-1864
15. Zadeh, L.A.: Fuzzy Logic Systems: Origin, Concepts, and Trends. Hong Kong, November 10 (2004)
16. Driankov, D., Hellendoorn, H., Reinfrank, M.: An Introduction to Fuzzy Control. Springer-Verlag, Berlin (1993)

Parameter Estimation of BACK Equation of Sate Via Gray-Box Neural Network Models

Ulises I. Bravo-Sánchez[1] *, Florianne Castillo-Borja[1], Ramiro Rico-Martínez[2], and Gustavo A. Iglesias-Silva[2]

[1] Departamento de Ing. Química, Instituto Tecnológico de Aguascalientes, Av. López Mateos 1801 Ote. Aguascalientes, Ags, México. ulisesbs@gmail.com

[2] Departamento de Ing. Química, Instituto Tecnológico de Celaya, Av. A. García Cubas s/n, Celaya, Gto., México

Abstract. Improvement of models for correct estimation of thermodynamic properties have been a topic of many contributions in chemical engineering. In general, parameter estimation is made with a priori formulation of their functionality. In this contribution we adjust model parameters with gray-box models based on neural networks, experimental data and a frame model to be improved. The BACK equation of state (EoS) in used to predict the compressibility factor from Pressure-Volume-Temperature data of n-alkanes. Also, we check the consistency of gray-box models in prediction of properties derived from BACK EoS. The advantage of this approximation is to complement the knowledge of a fundamental model with the capacity of neural networks for pattern recognition. The presence of a frame model guides the optimization of neural networks to find parameters ad-hoc to the model theory.

1 Introduction

Artificial neural networks (ANNs) have been proved to be an excellent tool to recognize data patterns presents, even non linear, in a set of data. However, in systems with many knowledge and research done, use of neural networks can be seen as a forward step in prediction but as a backward step in knowledge. The goal of this contribution is to show a gray-box model formulation based on a deterministic model, neural networks and a set of experimental data to improve the deterministic model. Once the gray-box model formulation has demonstrate a good performance in prediction and has been probed its consistency with fundamental principles, we could use the model with some confidence to extrapolate.

Research in thermo-physical properties of fluids and its modeling has been an area of intensive work. The most successful engineering thermodynamics equations of state are empirical, like Redlich-Wong, Peng-Robinson, Soave and more. In recent years several equation have emerged with a frame work based on fundamental principles. This kind of equations use a better molecular description or information of molecular dynamics. However, simplifications from theory are

* Thanks to: Conacyt SEP-2204-C01-47921

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 16–26

Received 18/06/07
Accepted 31/08/07
Final version 22/09/07

made when a model is constructed. Therefore, errors in prediction increased when the model is used in systems different from ones the model has been made. Then, the improvement of prediction of fluid properties still been an active topic in thermodynamics.

In this work, we chose the BACK EoS [6,7] as the frame model to be improved in its parameter functionality. Experimental PVT data for the family of n-alkanes from methane to n-decane are used to predict the compressibility factor (Z). BACK EoS consider two principal deviation effects from ideality, an effect product of repulsive forces and the other from attractive forces. The repulsive term relies in perturbation theory applied to convex bodies and the attractive term comes from an adjusted expansion in series to results of molecular simulation of Argon with a square-well potential. The convex bodies are allowed to have deviation from sphericity by mean of an anisotropy parameter (α), the two remaining parameters are a hard-core-volume related parameter ($\tilde{V}^0$) and the energy potential ($\frac{u^0}{k}$).

Neural networks can be used as a "black box" model. in this kind of model a data vector is feed to the neural net and its parameters are trained to recognize an output data vector. That kind of approximation only assures the best statistical set of neural network parameters to this task without insights of relationships more than existent theory. Extrapolate these models can not be successful at all and just serve to predict the properties of compounds used in training. On the contraries, a "gray-box" model uses the theory existent and just try to complement those fails, therefore the model can be more general. A strategy is necessary to achieve this goal. An iterative procedure is applied to make a diagnosis on which set of variables is the best candidate to drive the modeling without degenerate the frame model (BACK EoS). In the next sections, we present the BACK EoS and the details of the gray-box formulation and finally discussion and concluding remarks.

2 Equation of State Back

Among the vast number of equation of state, in recent years some EoS are emerged with a better fundamental description of fluids with its respective increase on complexity like BACK, PHCST, SAFT, PC-SAFT. The BACK EoS is not excessively complex and is oriented to alkanes; with its original parameters shows good agreement with experimental data until n-butane, and the average deviation increase as the number of carbons increase. Also, deviation from compressibility factor grows with density, that is a common fail in many EoS because are constructed around deviation from ideal gas ($\tilde{V} \to \infty$), so for specific volumes of liquid the errors are big.

Then, we have two facts that lead us to implement our strategy based on neural networks to improve the parameter estimation of an EoS previously constructed. The BACK as any EoS needs of temperature and specific volume besides its characteristics parameters to calculate the compressibility factor. The

BACK equation[6,7] is an augmented van der Waals EoS of the form:

$$\frac{PV}{RT} = Z = Z^h + Z^a \tag{1}$$

For the repulsive term (Z^h), the equation of Boublik is used:

$$Z^h = \frac{1 + (3\alpha - 2)\xi + (3\alpha^2 - 3\alpha + 1)\xi^2 - \alpha^2\xi^3}{(1-\xi)^3} \tag{2}$$

where α is the anisotropy parameter and ξ is the density of the fluid given by:

$$\xi = 0.74048\frac{\tilde{V}^0}{\tilde{V}} \tag{3}$$

Here $\tilde{V}$ is the specific volume and $\tilde{V}^0$ the molecular hard core volume. The Boublik equation is the result from apply perturbation theory on a fluid with molecules as convex bodies. For the attractive term (Z^a) we use the equation derived by Alder ([14]):

$$Z^a = \sum_{N=1}^{4}\sum_{M=1}^{9} MD_{NM}\left(\frac{u}{kT}\right)^N\left(\frac{\tilde{V}^0}{\tilde{V}}\right)^M \tag{4}$$

where T is the temperature, $(\frac{u}{k})$ is the potential energy and D_{NM} are universal constants.

3 Neural Network Gray-Box Models

The central part of succeed on this parameter estimation approach has to do with the use of the so called gray-box. Our definition of gray-box model involves a deterministic model and one or more neural networks, the topology or connectivity between those elements is defined by the purpose of correction. Once the topology is defined, we construct several gray-box models starting from a model with "minimal knowledge". That means we use the information available in data without take into account the model theory. Therefore, after training the model evolves incorporating information from theory and from observed of analysis to the model itself, to be trained again until the goal is reached.

The kind of neural networks used are feedforward with four layers, also knows as backpropagation neural networks. Each neuron, in feedforward networks, is connected with all the neurons in the previous layer via a weighted connection, which only pass information to all neurons in the next layer, avoiding lateral or recurrent connections. Also each neuron has a bias parameter that contribute to the neuron output. The problem to solve is to find the weights and bias of the neural networks. They are found by solving a least-squares problem using conjugate gradient techniques. Such ANN architecture is the most common and its configuration and training (solution of the least-squares problem) have been

described in detail elsewhere (e.g. [15]). The input layer just drives the PVT information to the neural network, two hidden layers functions as the non-linear part of the model and the output layer computes the predicted variables. In our case, the parameter(s) to be included in the fundamental model.

The models based on neural networks and a set of experimental data assures the patter recognition between the input-output vectors [19], but if we do not have an additional strategy then we can not extract relevant information of the system. On the contraries, a "gray-box" model uses the theory existent and just try to complement those fails, therefore the model can be extrapolated with more confidence. A strategy is necessary to achieve this goal. An iterative procedure is applied to make a diagnosis on which set of variables is the best candidate to drive the modeling without degenerate the frame model (BACK EoS).

The scheme of the first gray-box model (named ANN1 for short) is showed in Figure 1. The topology with three neural networks obey to avoid cross interaction among the selected dependency of each EoS parameters and the inputs to the nets. As mentioned previously, first we use information without consider the basis of BACK EoS, therefore in this model we use temperature, specific volume and the number of carbons as independent variables. One neural network uses the number of carbons to recognize the anisotropy factor, because this parameter is only function of molecular geometry and it is considered independent of temperature. In BACK's formulation, $\tilde{V}^0$ is considered temperature dependent. however in this model we use density as input information to the second ANN and finally for $\frac{u}{k}$ parameter the temperature is the independent variable. This

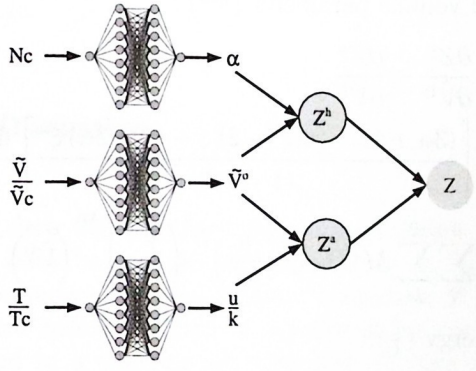


Fig. 1. Scheme of the first gray-box model, ANN1.

model was trained with data of methane, ethane and n-decane to observe if the corrections are consistent for all data. The model is training upon convergence on compressibility factor prediction (Z). The training is via minimization of the square error between the experimental compressibility factor and predicted

(energy function, E):

$$E = \sum_{np} \frac{1}{2}(Z_{exp}^{np} - Z_{pred}^{np})^2 \tag{5}$$

where np indexes the training vectors. The neural networks used have one neuron in the input layer (PVT information), one neuron in the output layer (BACK EoS parameter) and eight neurons in each hidden layer. The speed, final value of energy function and values of the final set of weights and thresholds are not as relevant as the values and behavior of the parameters of BACK EoS from the outputs neurons. The training is achieved using the *backpropagation rules*. Such rules are straightforwardly obtained using the chain rule to calculate the derivative of the energy measure with respect to the ANN parameters. For the first model, the derivatives of the energy function with respect of the neural networks parameters ($\boldsymbol{W}$ and $\boldsymbol{\theta}$) required for the backpropagation of the errors, have the general form:

$$\frac{dE}{dW} = -(Z_{exp} - Z_{pred}) \left(\frac{dZ}{dP} * \frac{dP}{dW} \right) \tag{6}$$

where P stands as an EoS parameter. The first part of the derivative $\frac{dZ}{dP}$ can be explicitly calculated from the BACK equation formulation and the second, $\frac{dP}{dW}$, is simply the derivative of one of the network outputs with respect to one of the network parameters. The derivative of compressibility factor with respect of the anisotropy parameter (α) is:

$$\frac{\partial Z}{\partial \alpha} = \frac{3\xi + (6\alpha - 3)\,\xi^2 - 2\alpha\xi^3}{(1-\xi)^3} \tag{7}$$

For the close-packed volume parameter ($\tilde{V^0}$) :

$$\begin{aligned} \frac{\partial Z}{\partial \tilde{V^0}} &= \frac{\partial Z^h}{\partial \tilde{V^0}} + \frac{\partial Z^a}{\partial \tilde{V^0}} \\ \frac{\partial Z^h}{\partial \tilde{V^0}} &= \left[\frac{(3\alpha + 1) + (6\alpha^2 - 2)\,\xi + (1 - 3\alpha)\,\xi^2}{(1-\xi)^4} \right] \frac{0.74078}{\tilde{V}} \\ \frac{\partial Z^a}{\partial \tilde{V^0}} &= \sum_{N}^{4} \sum_{M}^{9} M^2 D_{NM} \left(\frac{u}{kT} \right)^N \left(\frac{\tilde{V^0}}{\tilde{V}} \right)^M \left(\tilde{V^0} \right)^{-1} \end{aligned} \tag{8}$$

For the potential energy ($\frac{u}{k}$):

$$\frac{\partial Z}{\partial (\frac{u}{k})} = \sum_{N}^{4} \sum_{M}^{9} MND_{NM} \left(\frac{u}{kT} \right)^N \left(\frac{u}{k} \right)^{-1} \left(\frac{\tilde{V^0}}{\tilde{V}} \right)^M \tag{9}$$

A similar procedure is used for the remaining gray-box models. The second model presented in Figure 2, named ANN2, takes into account more information of BACK theory. We known that both parameters, V^0 and u/k, are function of temperature. Finally, in Table 1 we present the full range of conditions of experimental data used in training and cross validation.

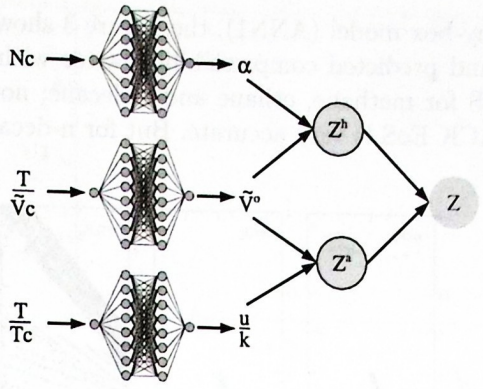


Fig. 2. Scheme of the second gray-box model, ANN2.

	T K	P bar	ρ $\frac{mol}{cm^3} \times 10^3$
CH_4	119.4 – 423.15	16.3 – 291.7	0.0239 – 35
C_2H_6	248.15 – 473.15	11.7 – 410.2	0.0313 – 16
C_3H_8	285.9 – 510.93	1.01 – 689	0.0240 – 13
C_4H_{10}	294.26 – 394.26	1.01 – 8.61	0.0102 – 3
C_5H_{12}	298.15 – 573.15	1.37 – 741	0.0240 – 9.72
C_7H_{16}	285.25 – 313.15	19.3 – 784	0.0670 – 7.95
C_8H_{18}	298.15 – 548	20.2 – 982	0.0352 – 7.48
$C_{10}H_{22}$	294 – 673	15.2 – 1500	0.95 – 4.8

Table 1. Range of pressure, density and temperature for the experimental database [16].

4 Results and Discussion

A set of 2000 PVT data from methane to n-decane, taken from [16], was the database used. All data belongs to one phase (gas or liquid) of pure compounds. Table 1 summarizes the experimental conditions of data. The data are distributed above and below the critical conditions for each compound. This database was divided in a training set with 80% of each compound set and the rest for cross validation. Every gray-box model was trained five times with random initial condition to validate the model. The results showed are representative of these trainings, the goal is the behavior observed not the final values of neural networks parameters or the number of conjugated gradient iterations needed to reach the final error. Even when the parameters of EoS are molecular properties and are knowns for some molecules, in common practice are considered as empirical parameters from a statistical fit of experimental data. Then, the fitting procedure absorbs the errors of model by idealities or simplifications.

For the first gray-box model (ANN1), the Figure 3 shows the comparison of the experimental and predicted compressibility factor with the model and the original BACK EoS for methane, ethane and n-decane; note that for methane and ethane the BACK EoS is very accurate. But for n-decane, were the predic-

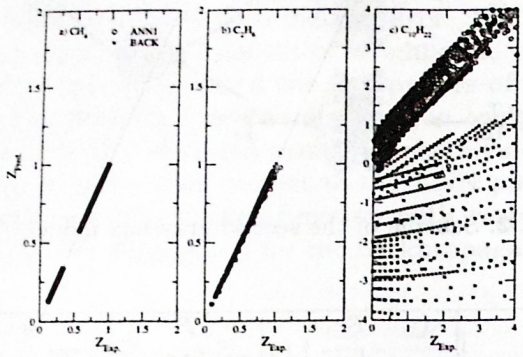


Fig. 3. Comparison of the experimental and predicted compressibility factor with the ANN1 gray-box model (○) and the original BACK EoS (×) for a) methane, b)ethane and c) n-decane.

tions are extremely worse, with almost every data, the predictions are improved significantly by the addition of neural networks. However, the analysis of the parameters obtained at the outputs of the neural networks is the indicator of succeed for the gray-box model. The anisotropy parameter (α) in Table 2 shows a value below but near unit for methane as was imposed in the original fitting. For ethane, when trained with one isotherm as methane, the α values are greater

	T, K	α_{ANN1}	α_{BACK}
CH_4	180	0.98690	1.0000
C_2H_6	273	0.99579	1.0370
	298	1.04466	
	323	1.02646	
	273, 298, 323	0.97474	
$C_{10}H_{22}$	294, 523, 653	3.29909	1.13490

Table 2. Predicted anisotropy parameter for several training sets with ANN1 gray-box model.

than α for methane; finally for n-decane the α value is too large to the "physical" meaning of the parameter. The close-packed volume parameter ($\tilde{V}^0$), as showed in Figure 4, behaves independent of temperature as the ANN1 model was constructed. all predictions converge to one function in density. However, the search

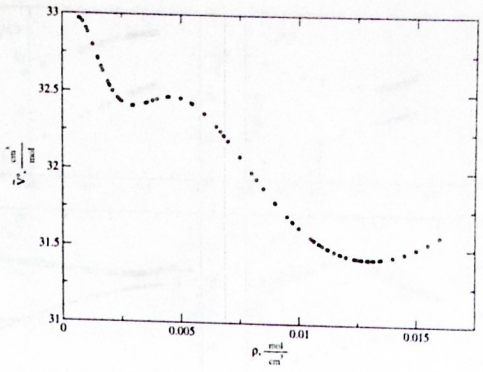


Fig. 4. Packed volume of convex bodies with the gray-box model ANN1 for ethane.

of a function to represent this behavior seems not to be a viable option. That observation comes from the frame of BACK EoS. if we add the density (volume) in the calculus of the close-packed volume parameter that implies a change in the structure of BACK EoS which can degenerate the calculus of another properties derived, such as entropy or enthalpy. For the potential energy ($\frac{u}{k}$) the behavior is smooth and similar to proposed in theory as function of temperature. For the second gray-box model (ANN2), a change in dependency of the second neural network in ANN1 model is made. then we get the ANN2 model. After training, the anisotropy parameter (Table 3) is consistent with the *principle of congruence* which increase its value as the number of carbons increase. However. is to note the high value for α of n-decane which is reflected in the behavior of the two remaining parameters. The close-packed volume parameter ($\tilde{V}^0$), as showed in

	α_{ANN2}	α_{BACK}
C_2H_6	0.9992124	1.037
C_3H_8	1.0164985	1.041
C_5H_{12}	1.0348073	1.0566
$C_{10}H_{22}$	6.128722	1.1349

Table 3. Predicted anisotropy parameter for ANN2 gray-box model.

Figure 5. behaves consistent from methane and n-propane with a tendency to decrease with temperature. From n-pentane the dependency from temperature shows a maxima and for n-decane we found negative values which are "physically" inconsistent. Also, for the potential energy ($\frac{u}{k}$) the behavior is consistent from methane and n-propane. for n-pentane some irregularities are presented and for n-decane, positive values are found but the dependence with temperature is inverted with respect BACK EoS theory. Finally we present, as a consistency test of the ANN2 gray-box model, the calculus of the vapor pressure from BACK

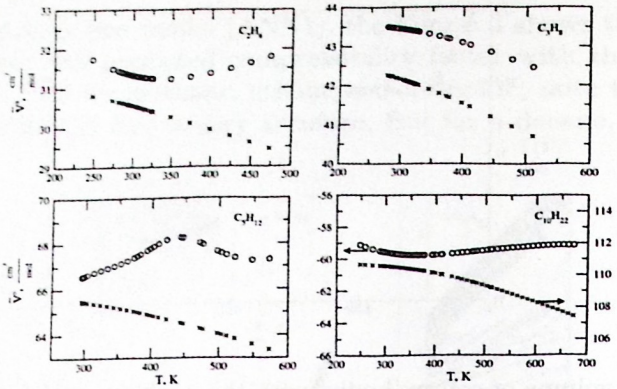


Fig. 5. Comparison of the packed volume of convex bodies with the ANN2 gray-box model (○) and the original BACK EoS (×) for ethane, propane, n-pentane and n-decane.

EoS with the Maxwell's rule of equal areas. This method implies that:

$$P(\tilde{V}^{vap} - \tilde{V}^{liq}) = \int_{\tilde{V}_{liq}}^{\tilde{V}_{vap}} P_{BACK} d\tilde{V} \tag{10}$$

In Figure 7 we show the agreement of the ANN2 gray-box model to predict the vapor pressure, with some discrepancies for n-decane. Note, the tendency is captured by the model but as the number of carbons increased the errors grows too.

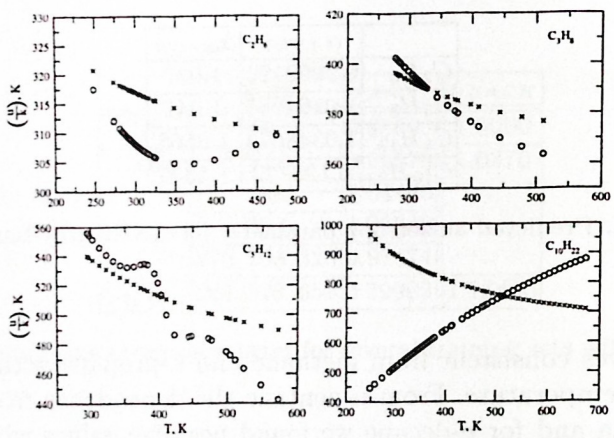


Fig. 6. Comparison of the potential energy with the ANN2 gray-box model (○) and the original BACK EoS (×) for ethane, propane, n-pentane and n-decane.

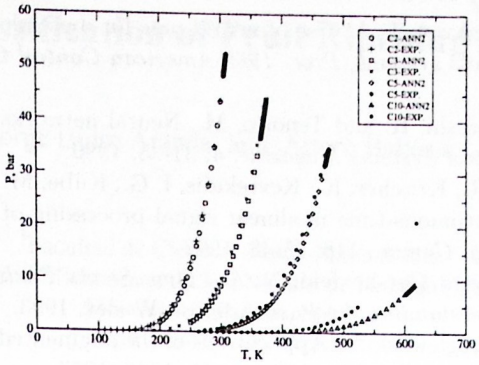


Fig. 7. Comparison of vapor pressure predicted with ANN2 model and experimental data.

5 Summary and Conclusions

We had shown an extended application of gray-box neural network in diagnosis and improvement of BACK EoS prediction of compressibility factor and some insight in its availability to predict derived properties. The strategy followed shows how discriminate a viable set of system variables to fit the parameters of BACK EoS. Additional effort is necessary to substitute the neural(s) network(s) with an empirical compact form that fits the neural networks behavior. This could leads to a better deterministic model with some empirical corrections derived from "what the models needs not what we want to add". However, some issues of this approach has to be mentioned. The optimization of the model is unconstrained and some rigid assumptions of model parameters can not been assured from the beginning. In fact, the three parameters must be always positive and in our approach that condition was suggested to follow by means of scaling the inputs and outputs in some expected range of values. Currently, we are conducting several related and complementary efforts. We are working in refining the models here presented with some insights of improvement. Also, we are looking into the formulation of corrections for more complex equations of state like SAFT for associating compounds (alcohols), where four parameters are needed by the equation. Also, adding constraints in this kind of gray-box models because many physical systems are subjects to constraints.

Acknowledgements

This work has been partially funded by the Mexican Ministry of Public Education(CONACyT: Conacyt SEP-2204-C01-47921 and DGEST).

References

1. A. Bulsari (ed.), *Neural Networks for Chemical Engineers*, Elsevier Science B. V. (1995).

2. Bhat, N. and McAvoy, T. J., Use of neural nets for dynamic modeling and control of chemical process systems, *Proc. 1989 American Control Conference*, 1342-1347, 1989.
3. Chu S. R., Shoureshi, R. and Tenorio, M., Neural networks for system identification, *IEEE Control Systems Magazine*, 4, 31-35, 1990.
4. Rico-Martínez, R., Krischer, K., Kevrekidis, I. G., Kube, M. C. and Hudson, J. L., Discrete- vs. continuous-time nonlinear signal processing of Cu electrodissolution data, *Chem. Eng. Comm.*, 118, 25-48, 1992.
5. Weigend, A. S. and Gershenfeld, N. A., *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.
6. Chen, S.S and Kreglewski, A., Applications of the augmented van der Waals theory of fluids, *Ber. Bunsenges. Phys. Chem.*, 81, 1048, 1977.
7. Kreglewski, A., *Equilibrium Properties of Fluids and Fluid Mixtures*, Texas A&M University Press, College Station, Texas, 1984.
8. Normandin A., Grandjean, B. P. A. and Thibault, J., PVT data analysis using neural network models, *Ind. Eng. Chem. Res*, 32, 970-975, 1993.
9. Habiballah W. A., Startzman R. and Barrufet, M. A., Use of neural networks for prediction of vapor-liquid equilibrium K-values for light hydrocarbon mixtures". *SPE Reservoir Engineering*, 121-126, 1996.
10. Sharma, R., Singhal, D., Ghosh, R. and Dwivedi A., Potential applications of artificial neural networks to thermodynamics: vapor-liquid equilibrium predictions. *Comput. Chem. Engng.*, 23, 385-390, 1999.
11. Psichogios D. C. and Ungar, L. H., A hybrid neural network-first principles approach to process modeling, *AIChE J.*, 38, 1499-1511, 1992.
12. Rico-Martínez, R., Anderson, J. S. and Kevrekidis, I. G., Continuous-time Nonlinear Signal Processing: A Neural Network Based Approach for Gray Box Identification, *Proceedings of the 1994 IEEE Workshop on Neural Networks for Signal Processing*, IEEE Publications, 596-605, 1994.
13. Petersen, R., Fredenslund, A. E. and Rasmussen, P., Artificial neural networks as a predictive tool for vapor-liquid equilibrium, *Comput. Chem. Engng.*, 18, S63-S67. 1994.
14. Alder B.J., Young D.A., and Mark M.A., *J. Chem. Phys.*, 56, 3013, 1972.
15. Lapedes, A.S., and Farber, R., Nonlinear signal processing using neural networks: Prediction and system modeling, *Los Alamos Report* **LA-UR 87-2662**, 1987.
16. GERG Technical Monograph, *The GERG databank of high accuracy compressibility factor measurements*. Technical Monograph No. 4, 1990.
17. Simnick, J.J., Lin, H. M. and Chu, C. K., The BACK equation of state and phase equilibria in pure fluids and mixtures, *Equations of State in Engineering and Research*, Advances in chemistry Series, American Chemical Society, 182, 209-233, 1979.
18. Avlonitis, D., Danesh, A., and Todd, A.C., Prediction of VL and VLL equilibria mixtures containing petroleum reservoir fluids and methanol with a cubic EoS, *Fluid Phase Equilibria*, 94, 181-215, 1994.
19. Cybenko, G., Approximation by superpositions of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, 2, 303-314, 1989.

Implementation of Data Fusion in a Probabilistic Neural Network (PNN) for Classification of Fruit Ripening Stages

Jorge Isidro Aranda[1] and Arturo Baltazar[2]

[1]Facultad de Ciencias Físico-Matemáticas
Universidad Michoacana de San Nicolás de Hidalgo (UMSNH)
Morelia, Michoacán, México
jorge_isidro@yahoo.com

[2]Centro de Investigación y Estudios Avanzados
CINVESTAV-Saltillo
Robotics and Advanced Manufacturing Program
Km. 13.5 Ramos Arizpe, Coahuila, 25900, México
arturo.baltazar@cinvestav.edu.mx

Abstract. Quality control of postharvest fruits is moving to substitute traditional sensory testing methods for more reliable quantitative methods. Ripening in fruits, such as tomatoes, is a complex phenomenon which affects chemical and physiological properties as function of time. Attempts to solve the problem of ripening classification have been focus on single sensors; however, there is not yet a complete solution. In this work, data fusion from different non-destructive sensors implemented through a probabilistic neural network is used to improve the quality control of ripening in tomato fruit. Two independent sensors, a novel non-destructive acoustic impact and a colorimeter technique were used. The effect of the probabilistic neural network parameters was explained by an analysis of Bayes error. The acceptance rate as function of the number of feature is analyzed. The results showed that the error classification rate was reduced from 60% to 10% using the proposed data fusion scheme.

1 Introduction

The goal in quality control of postharvest fruits has been to substitute traditional sensory testing methods for more reliable and quantitative methods by correlating quality properties with properties that can be quantitatively measured. Firmness has been used by several researchers to describe internal and superficial properties and it is considered an important quality characteristic of fresh postharvested tomato fruits, whole or sliced [1]. Color change is another important parameter affected by ripening in tomatoes. It is influenced by many factors, including: temperature, maturity stage, atmosphere and storage time. Tomato color is sometimes determined using sensorial

© A. Gelbukh, Á. Kuri (Eds.) Received 04/06/07

methods; however, instrumental methods using colorimeter provide quantitative and more effective ways to determine a color index. Ripening in fruits is a complex phenomenon which affects chemical and physiological properties as function of time with global and local variations than can be better described in probabilistic terms.
Measurements of resonance frequencies and their relation with elastic properties in fresh commodities with like-spherical or spheroidal shape were first described by Abbott et al. [1], and the formulation was later modified by Cooke [2]. Several techniques have been studied in the past to excite the resonance frequencies in fruits including force vibration [1] and acoustic impact tests (introduced by Yamamoto Iwamoto & Haginuma [3]). The relation between the resonance frequency and the elastic properties of fruit with spherical shape is given by

$$Sc = f^2 * m^{2/3}, \tag{1}$$

where Sc is typically referred to as stiffness coefficient, f is the dominant frequency and m is the bulk mass of the fruit. The determination of the dominant frequency f can be obtained by the acoustic impact test which consists of impacting the fruit with a small spherical object and recording the resonance vibration signals to determine the frequency spectrum [4]. In like-spherical fruits such as tomatoes two main types of vibration modes, assuming free boundary conditions, can be found: spherical modes and torsional modes. Some modes such as torsional can be filtered out using microphones, which are only sensitive to spherical vibration modes. However, a large uncertainty in the stiffness coefficient measurements can be found [4]. Although under controlled laboratory conditions they have been consistent and controllable results there is still presence of uncontrollable sources of noise [4].
The integration of sensor data using data fusion can be either complementary or use to enhance the response of a single sensor. In the area of robotics sensor fusion has been thoroughly investigated and used to overcome limitations of individual sensors [5]. Several schemes have been proposed to effectively develop and assess a methodology to integrate data [6]. The use of data fusion in robotics spans from improving position location and distance assessment to pattern recognition combining information from different sensors [5]. Artificial neural networks can be used for this fusion, and statistical models with probabilities help to integrate data at the evidential level, this level makes use of Bayesian approach theory.
Attempts to solve the problem of ripening classification have been mostly focused on single sensors; however, there is not yet a complete solution. Recently, application of artificial intelligence has been studied for sorting of fruits (i.e. [7] and [8]). Sensor fusion methodology has been applied using different non-destructive techniques including image analysis, near-infrared spectrophotometer, and colorimeter, among others. In this work, data fusion from different non-destructive sensors is used to improve the quality control of ripening in tomato fruit. The importance of data fusion of different sensors in foods such as melons and peaches to asses quality [8]. Sensor characteristics are a limiting factor in the performance of a data fusion system and good sensors are needed to obtain reliable results.
In this work, data integration and a fusion scheme is used to gather information of a proposed sensory system to produce a more complete solution to the ripening classification problem. Two independent sensors, a novel non-destructive acoustic impact [4] and a colorimeter technique [9] provided three different features: firmness,

color and luminosity. A PNN is used for data fusion and results are analyzed by a Bayesian classifier.

2 Probabilistic Neural Networks

Probabilistic Neural Networks (PNN) substitute the sigmoidal activation function commonly found in neural networks by an exponential function. Therefore, PNN results in a neural network that could compute nonlinear decision boundaries and approach the Bayes optimal [10]. PNN learns from a training data set, which is a collection of examples previously identified, belonging to specified classes. This training data are vectors with a determined number of components or elements each of them corresponding to the same number of features, each feature is obtained from a determined sensor.

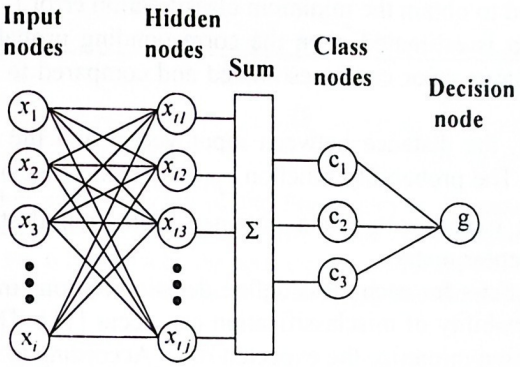


Figure 1. Probabilistic Neural Network

As described in Figure 1, a typical arrangement for a PPN is composed of two layers. The first layer contains input nodes where each receives an input value from the corresponding element in the input vector x and produces vectors indicating how close the input vector x is to the known pattern. This input vector data, labeled x, is a vector with the same number of elements or features as the training vector data. The second layer produces a vector of probabilities which is the sum for each class contribution in the first layer. Finally a hidden node with an activation function is given by

$$g = \exp(-\frac{D_i^2}{\sigma_i}), \tag{2}$$

where D_i is the distance between input vector and the i^{th} is the training data. This function g is a Gaussian function which returns a value of 1 if the input vector x and training data are equal, and drops to an insignificant value as the distance D_i increases.

Each hidden node is connected to a single class node C_i; if the output class of the input vector x is j, then such node is connected to the j^{th} class. Each *j-th* class node computes the sum of the activations (using Eq. 2) of the hidden nodes that are

connected to it (i.e. all the hidden nodes for a particular class) and passes this sum to a decisión node, which outputs the class with the highest summed activation. The output is the class that x seems most likely to belong to.

It is important to note that we are assuming that σ_i has the same value for each data, $\sigma_i = \sigma$. The sum of the activations for data belonging to a certain class gives rise to a probabilistic function (linear normalizad combination of the Gaussians centered in each of the data in that class) and computes the probability of the input vector belonging to that class. The probabilistic function is written as [10]

$$p(x|w_i) = \frac{1}{(2\pi\sigma^2)^{3/2} M_i} \sum_{j=1}^{M_i} \exp\left[\frac{-(x-x_{ij})^t (x-x_{ij})}{2\sigma^2}\right], \tag{3}$$

where i (=1, 2, 3) is the corresponding class number, j (=1,..., M_i) is the pattern number, x_{ij} is the j[th] training vector from class i, x is the test vector, M_i is the number of training vector in class i, σ is a smoothing factor, which is a variable that needs to be optimized to obtain the minimum classification error.

Once the optimum σ is estimated, then the corresponding probability functions for each class and the Bayes error can be estimated and compared to error classification obtained in the PNN.

In the present work, the distance between input vector and the i^{th} training data is $D_i = (x-x_{ij})^t (x-x_{ij})$. The probability function $p(x|w_i)$ (using Eq. 3) for each class was estimated using the experimental data obtained from non destructive techniques acoustic impact and colorimetry .

The probability functions for each class define decision regions in which boundaries with the highest probability of misclassification can occur [11]. The Bayes strategies for pattern classification minimize the expected risk. According to the Bayes decision rule to assign a pattern x to a class w_i with minium error, we have that

$$p(w_i)p(x|w_i) > p(w_k)p(x|w_k) \quad k,i = 1,...,n; k \neq i, \tag{4}$$

where $p(w_1),..., p(w_n)$,are known prior probabilities [11]. The average probability of error for the case of three classes is given in terms of decision regions in which vector x falls

$$p(error) = \int_{R_1} p(w_1)p(x|w_1)dx + \int_{R_2} p(w_2)p(x|w_2)dx + \int_{R_3} p(w_3)p(x|w_3)dx, \tag{5}$$

which is known as Bayes risk [12] and x satisfies the following equation $p(w_i)p(x|w_i) > \max_{\substack{1\le k\le 3 \\ k\neq i}} \{p(w_k)p(x|w_k)\}$ with this the Bayes risk becomes the Bayes error, considering Eq. 4. The exact calculation of Eq. 5 for multi-class and multivariate problems is quite difficult, even assuming normal distribution [11]. This is due to the discontinuous nature of the decision regions. If the intersection area between classes 1 and 3 vanishes, then the three-class problem can be approached as a two-class problem and only the first two terms in Eq. 5 are needed [13].

However, Eq. 5 can be approximated analytically to give an upper and lower bound on the error [11]. These can be estimated in terms of the Bhattacharyya bound, which is a particular case of Chernoff bounds as follows

$$p(error) \equiv \varepsilon \leq \sqrt{p(w_1)p(w_2)} \int \sqrt{p(x|w_1)p(x|w_2)}\, dx. \tag{6}$$

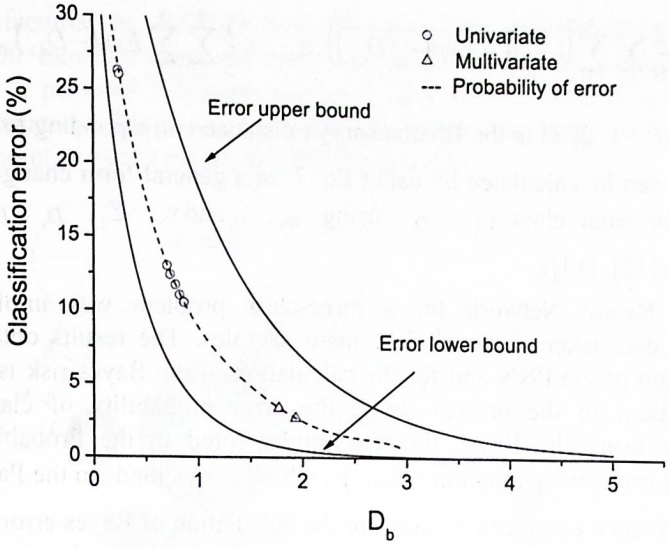


Figure 2. Classification error in terms of the Bhattacharyya distance. Solid lines are the theoretical classification error bounds. The estimate of the classification error for data in the Univariate case using Eq. 6 is shown by the open circles and for the Multivariate case by open triangles. The theoretical probability of error as given by Eq. 8 is shown by the dashed line

However, Eq. 5 can be approximated analytically to give an upper and lower bound on the error ([14]). Chulhee and Choi [14] presented an approach to estimate the classification error ε in the case of two classes (in this case $p(w_1) = p(w_2) = 1/2$), which reduces to the probability error for low values of Bhattacharyya distance D_b [12], [13], [14], given by

$$D_b = \frac{1}{8}(\mu_2 - \mu_1)^t \left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}, \tag{7}$$

where μ_l and Σ_l (l = 1,2) are the mean and covariance for each multivariate class. The approach uses a fitted polynomial curve obtained using different class statistics. The polynomial curve is described as [14]

$$E = 40.219 - (70.019)\, D_b + (63.578) D_b^2 - (32.766)\, D_b^3 + (8.7172) D_b^4 - (0.91875) D_b^5, \tag{8}$$

this equation is valid only for $D_b \leq 3$ (see Figure 2) and shows t n the number of features increases, the classification error diminishes significa.

For a three-class problem, Garber and Djouadi [12] introduced a technique to estimate bounds for the classification error (ε); such bounds are expressed as a function of pairwise classification errors which for a three-class problem is evaluated by adding

the probability error resulting in three combinations of pairs of classes. The classification error for deciding among the original three classes ε is bounded from above and below as ([12])

$$\varepsilon_{min} \geq \frac{2}{9}\sum_{i=1}^{2}\sum_{j=i+1}^{3}\left(1-\sqrt{1-Exp(-2D_{i,j})}\right), \varepsilon_{max} \leq \frac{1}{3}\sum_{i=1}^{2}\sum_{j=i+1}^{3}Exp(-D_{i,j}) \,, \tag{9}$$

where $D_{i,j}$ $(i, j = 1, 2, 3)$ is the Bhattacharyya distance corresponding to the events w_i and w_j, each can be calculated by using Eq. 7 in a general form changing mean and covariance for each class ($D_b = D_{3,1}$ using μ_3, μ_1 and Σ_3, Σ_1; $D_b = D_{3,2}$ using μ_3, μ_2 and Σ_3, Σ_2) ([12], [14]).

Probabilistic Neural Network for a three-class problem was implemented for experimental data taken from whole tomato samples. The results obtained for the implementacion of the PNN and for the calculation of the Bayes risk is discussed in the next section. In the present work, the error probability of classification is calculated by using the Bayes theorem implemented in the Probabilistic Neural Network. The probability function $p(x|w_i)$, which corresponds to the Parzen window in the PNN for each class, can be used for the calculation of Bayes error by using Eq. 5; the Bayes error becomes Bayes risk [12] when the PNN is at the stage of decision node.

3 Experimental Description

In this section, materials and methods for the classification of tomatoes are described in detail. Whole tomato fruit samples, Charleston ("Lycopersicum esculentum"), were obtained from a greenhouse located in Sonora (México). The tomatoes were harvested and preliminarily sorted with colorimeter leaving only those with roughly Breaker color (USDA classification). A total of 236 whole tomato samples were used in the experiment. The experimental considered a storage temperature of 20 °C. The relative humidity of the cold chamber was 75%. Colorimeter, weight loss and nondestructive firmness measurements were taken for these samples at selected testing days.

Color measurements were performed using a Minolta CR 300 colorimeter; three replicas were taken for each intact tomato sample. The equipment provides an estimate of values L*, a*, and b* recommended by CIE [9] from where a*/b* ratio, chroma and hue angle = tan-1(a*/b*) can be calculated. Measurements of weight loss were done using a laboratory weighing instrument, Mettler Toledo model PR2003 Deltarange ® with readability of 1 mg. The impact acoustic tests were carried out using the experimental setup described in Figure 3. The specimen was insulated from external vibrations or noise using a simple plastic foam layer located underneath it, allowing the sample to vibrate freely. The impacting force was generated by a mechanical pendulum consisting of a small solid plastic sphere with a 24.9 mm diameter, 3.09 grms in weight and a string of 180 mm in length. The impacting sphere

was released from an angle of θ =30 degrees as shown in Figure 3. Acoustic signals were recorded at two points: 0 degrees and 90 degrees in relation to the impact direction of the pendulum. The signals were captured by a sound sensor model CI-

6506B manufactured by PASCO Scientific and then digitalized by the Science Workshop 500 Interface from the same manufacturer. Finally, signals were post-processed by a personal computer equipped with FFT to determine the dominant resonance frequency. Three repetitions of impact acoustic tests were performed at each impact point to total 6 measurements per sample.

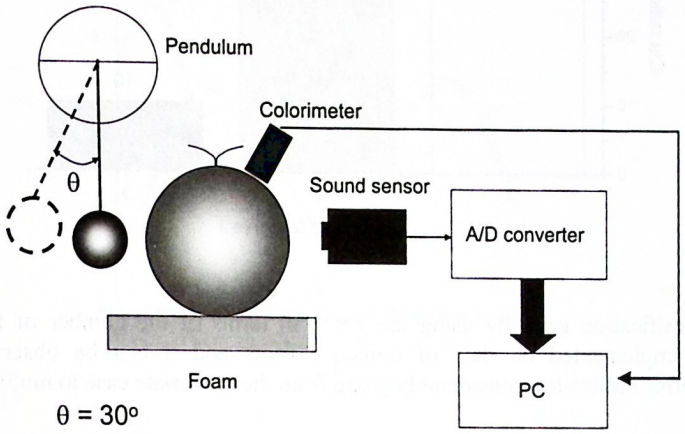


Figure 3. Experiment setup for impact acoustic and colorimeter tests on tomato samples

Three classes were chosen for training of the PNN, a total of 239 data vectors were considered for classification analysis. Data was sorted into three classes based on time in storage, Class 1: from days 0 to 6, class 2: days 6 to 15 and class 3: days 15 to 27. Then, 45 neurons (with known class) were randomly chosen for training data in the input layer. The features were put in vectors as follows: [a*/b*, Sc, L*]. Matlab 7.0 ® was used for the design and construction of the PNN.

4 Results

Tomato samples at a temperature of 20^0C degrees were studied. Data taken from the acoustic impact and colorimeter sensors for the tomato samples was presorted into two sets: training data and input data vectors [a*/b*; Sc; L*]. The data set consisted of 239 data vectors taken from samples at 20^0C; 45 vector data were used for training the network and 194 for classification. The optimum σ value for the three features (a*/b*, Sc, L*) with the best classification results was found at σ = 0.1. It is important to note that when only one feature is considered the corresponding error classification is about 60%. This is better observed in Figure 4, where a reduction in error classification from 60% to 10% as a function of number of features was estimated using the proposed PNN.

It can be also observed, that the results in Figure 5 agree with those in Figure 4.

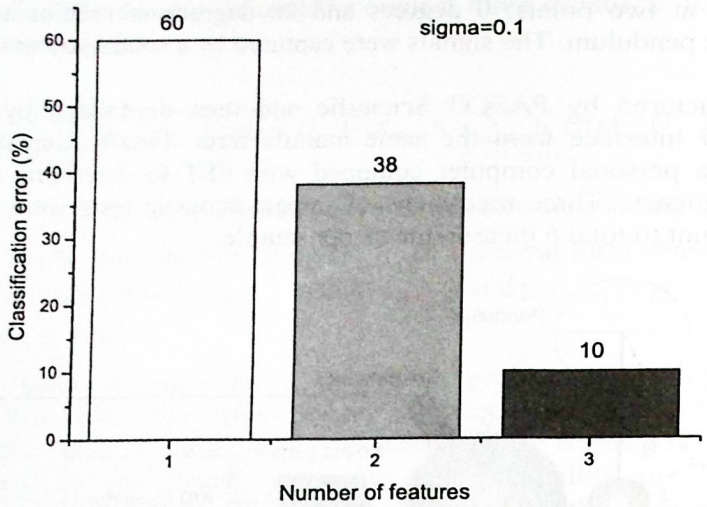


Figure 4. Classification error by using the PNN, in terms of the number of features. The network was implemented on data of tomatoes fruits and it can be observed that the classification error diminishes considerably going from the univariate case to multivariate case

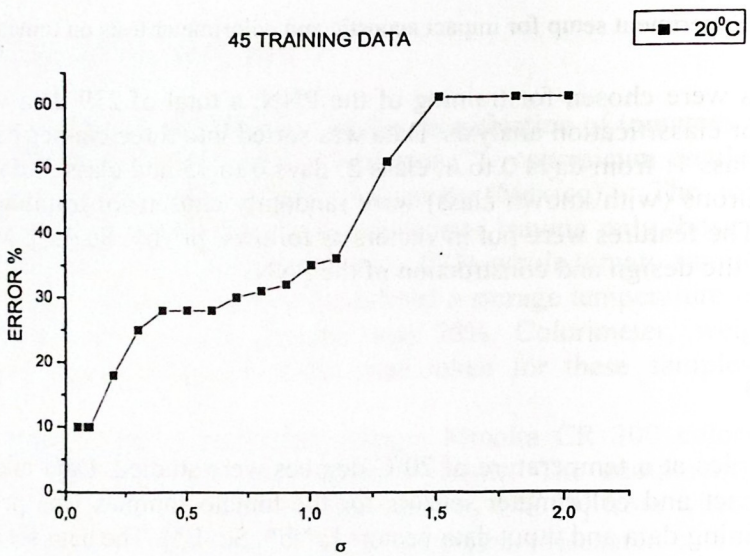
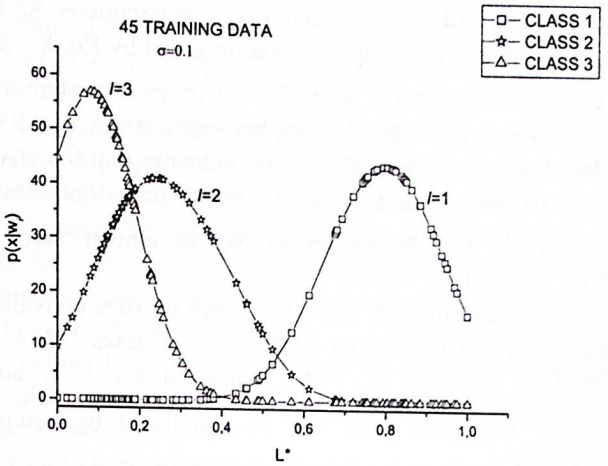
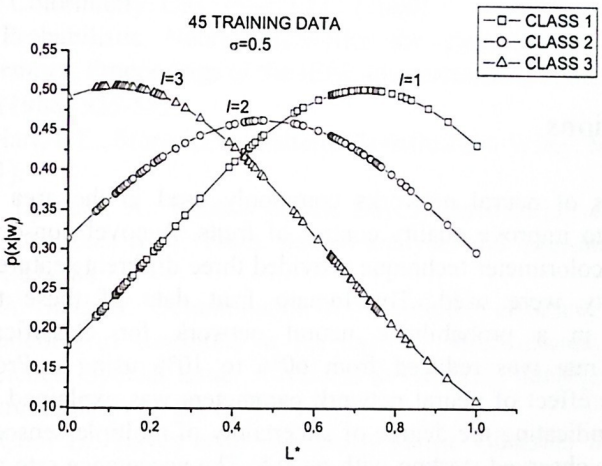


Figure 5. Results for the classification error in function of σ for data belonging to three classes and three features (a*/b*, Sc and L*). 45 vector data at 20^0 C were used for training and 239 input data vector for classification for tomato samples at 20^0 C were considered. The minimum error was obtained with a value of σ=0.1

The classification error in terms of the smoothing parameter σ in the PNN design is shown Figure 5 which was estimated as the number of times that the network incorrectly classified a known sample divided by the total number of input data. The results are the fusion of the three vector features corresponding to two sensors.
The probability functions for the multivariate multiclass case were obtained for each class and compared with the Bayes error classification calculated for $\sigma = 0.1$ (Eq. 9).



a)



b)

Figure 6. Probability functions corresponding to color parameter L* where data correspond are at 20^0 C. 45 training vector data and 239 input vector data, at 20^0 C, for classification were used for all figures with probability functions for three classes (*class 1 for square points, class 2 for circle points, class 3 for triangle points*). The figures correspond to different values of σ . a) $\sigma = 0.1$; b) $\sigma = 0.5$

The intersection between the corresponding curves probability functions for L* (luminosity) is more significant as the value of the σ increases (Figure 6).
The calculation of the Bhattacharyya distance and the classification error for the univariate and multivariate cases indicates that in the univariate case, Bhattacharyya distance between classes 1 and 3 is larger than 3.0. Therefore, bounds in Eq. 6 are valid to estimate the mean classification error; same criteria was used for classes 1-2 and 2-3 as defined for parameter L*. The classification error for classes 1-2 and 2-3 were estimated as 11% and 12% respectively. For parameter Sc the Bhattacharyya distance is $D_b = 0.7$, therefore, the approximation given by Eq. 8 was used; the results show that the classification error is about 13%. Curves of probability functions of L* feature (Figure 6) show a clear interference between classes 1 and 3. However, this is reduced after data fusion (multivariate) which indicates that the error between classes 1 and 3 is close to zero, $\varepsilon_{3,1} \approx 0$, here it can be seen that classification error is diminished from 13%, for the parameter Sc, to almost zero considering three parameters (a*/b*, Sc, L*).
The total error in the classification gives the result of 10% considering Eq. 9, in this case the corresponding Bhattacharyya distances for classes 1-2, 1-3 and 2-3 are $D_{2,1}$ =1.93 (using Eq. 8 $\varepsilon_{2,1} = 2.6\%$), $D_{3,1}$ =18.9 (using Eq. 8 $\varepsilon_{3,1} \approx 0$) and D_b =1.77 (using Eq. 8 $\varepsilon_{3,2} = 3.3\%$). This total error can be calculated by using Eq. 10 and the corresponding classification errors calculated for pairs of classes. Classification error obtained by PNN for two sensors and three features is in agreement with results of Bayes risk. This consideration can stand for three features and σ = 0.1 as discussed above.

5 Conclusions

Methodologies of neural networks commonly used in the area of robotics were implemented to improve quality control of fruits. A novel non-destructive acoustic impact and a colorimeter technique provided three different features: firmness, color and luminosity were used. The tomato fruit data of these two sensors was implemented in a probabilistic neural network for classification. The error classification rate was reduced from 60% to 10% using a Probabilistic Neural Network. The effect of neural network parameters was explained by an analysis of Bayes error, indicating the degree of uncertainty of multiple sensors during ripening stages. This is observed starting with σ= 0.5. The acceptance rate results as function of feature factors was analyzed. The results found in tomato fruits can be potentially extrapolated to other postharvest fruits.

References

1. Abbott, J., R. Lu, B.L., & Upchurch R.L.: Technologies for Nondestructive Quality of fruits and vegetables. Horticultural Reviews, 20, (1997) 1-120.
2. Cooke J.: An interpretation of the resonant behavior of intact fruits and vegetables. Transactions of the ASAE, 15(6), (1972) 1075-1080.
3. Yamamoto, H., Iwamoto, M. Haginuma, S.: Acoustic impulse response method for measuring natural frequency of intact fruits and preliminary applications to internal quality evaluation of apples and watermelons. Journal of Texture Studies, 11, (1980) 117-136.
4. Baltazar, A., J. Espina, I. Ramos-Torres and G. Gonzalez-Aguilar.: Effect of Methyl Jasmonate on properties of intact tomato fruit monitored with destructive and nondestructive tests. Journal of Food Engineering, 80 (4), (2007) 1086-1095.
5. Stephen S.S.: Synesthetic sensor fusion via a cross-wired artificial neural network, thesis. East Tennessee State University, USA (2002).
6. Kokar M., Tomasik J., et al: Formalizing classes of information fusion systems. Information Fusion 5: (2004) 189-202.
7. Shahin M. A., Tollner E. W., McCLendon R. W. Artificial intelligence classifiers for sorting apples based on Watercore. Journal of Agricultural Engineering Research, 79(3), (2001) 265-274.
8. Steinmetz V., Sévila F. and Bellon-Maurel V.: A methodology for sensor fusion design: Application to fruit quality assessment. Journal of Agricultural Engineering Research, 74(1), (1999) 21-31.
9. Marcus, R.T.: Colorimetry. CRC Press LLC. (2000).
10. Specht D.: Probabilistic Neural Networks for classification, mapping or associative memory. Proceedings of the IEEE International Conference on Neural Networks, 1, (1988) 525-532.
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, Second Edition. (2001).
12. Garber, F.D., Djouadi, A.: Bounds on the Bayes classification error based on pairwise risk functions. IEEE Trans. pattern analysis and machine intelligence. 10(2), (1988) 281-288.
13. Aranda, J. I., A. Baltazar.: Determination of the optimum error and reject validity regions as function of threshold in the case of three classes. Research in computing Science. Special issue: Advances in Artificial intelligence, Gelbukh and Reyes-García Eds., Vol. 26, (2006) 61-72.
14. Chulhee L., Choi E.: Bayes error evaluation of the Gaussian ML Classifier. IEEE: Transactions on geoscience and remote sensing, Vol. 38, No. 3, (2000) 1471-1475.

Flexural Strength Prediction of Slabs with Finished Type Tile: A Neural Network Approach

J. Acevedo-Dávila and L. Torres-Treviño

Corporación Mexicana de Investigación en Materiales
Oceania 190, Saltillo, Coahuila, México
ltorres@comimsa.com.mx; jacevedo@comimsa.com.mx

Abstract. An application of neural network has been applied to the development of materials for the construction industry. Different mixes were designed to create a piece which will be used on a roof with an angle of inclination and some tile. The experiments were divided in two stages. The first consisted of the development of mixes with different amounts of cement. The flexural strength was evaluated. In the second stage, a Percetron neural network based on experimental tests was generated and evaluated in order to predict mechanical properties of new mixes. This study shows the correlation of the different mixes and the neural network prediction as well as the behavior of other designed mixes.

1 Introduction

The lack of a worthy ceiling in the housing industry in Mexico is one of the most serious problems that is presented to the population with low resources. According to data emitted by INEGI from XII population Census [1], 35 percent of the population in the country has ceilings made of material classified as low durable. That numbers nearly 7.6 million houses. According to INEGI, these materials, shown in table 1, show a nominal life utility that go from 2 up to 50 years.

Table 1. Font Life utility of the different materials used for the construction of ceilings

Material	Nominal Life utility (years)
Waste material	2
Cardboards	2
Metallic and asbesto	11
Palms, tejamanil and wood [2, 3]	15
Tile [2,3]	30
Concrete slabs, bricks and terrace with a set of beams	50

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 38–44

Received 16/06/07
Accepted 31/08/07
Final version 30/09/07

An area of opportunity in the housing industry in Mexico, based on the aforementioned problem, is the development of constructive systems for lasting ceilings or cover according to construction norms [4]. COMIMSA has developed a piece which will be used on a roof with an angle of inclination and some tile [5]

The system consists basically of two pieces of slabs of concrete lifting elements and, concrete slabs with finished type tile. The applied process of vibration and compacting during the manufacture of blocks is done in order to increase the volumes of manufacture of concrete slabs proposed [6]. An alternative to improve the properties of resistance to flexion is the increase of cement. Nevertheless, this raw material is expensive, which makes it indispensable to look for methodologies that help to optimize the used cement and to make an experiment design that diminishes the amount of necessary experiments.

Flexural strength is the ability to resist an applied bending force such as that encountered by concrete pavements or other slabs. A determination of the flexural strength is frequently necessary as part of the design of concrete mixtures in order to check compliance with established specifications or to provide information necessary to the design of an engineering structure. In the flexural-strength test, a test load is applied to the sides of a test beam. Although the test can be performed upon beams sawed from existing concrete structures, it is more commonly performed upon beams that are cast for testing purposes.

The use of neural networks is an alterative to make economic designs by the capabilities prediction of these computational paradigms [7, 8, 9, 10, 11, 12]. The principal use of this paradigm is the approximation of functions and processes using experimental data. There are several approaches to neural networks and several learning rules in a single neural structure such as: gradient with momentum, conjugates, gradients, etc. Nevertheless a erceptron neural network with backpropagation Levenberg - Marquardt learning rule is used.

In following section an overview of Perceptron neural networks is given. The third section discusses the experimental procedure to extract the training and validation data. Results are given in section four. Finally, conclusions and future work will be given in the last section.

2 Perceptron Neural Network and the Leverberg - Maquardt Learning Rule

A neural network is an interconnection of processing elements or artificial neurons [13]. The principal use of this paradigm is the approximation of functions using experimental data. The behavior of the neural network is established by its' connection weights. The learning rule adjusts the weights to minimize any error between a desired response and the actual response of the neural network. The desired responses are expressed in a database of patterns of input – output pairs. A single response of a processing element is determined with (1) and (2).

$$O_n = F_A(\sum_{i=1}^{N} W_{n,i} X_i) \tag{1}$$

$$F_A(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Where n is an artificial neuron and i is the corresponding element of an input vector X. Matrix W represents the connection weights between an element of the input vector X and a neural element n. The activation function F_A regulates the output of the neuron. A Sigmoid function is usually used (2). Since layers of neurons are used, it is necessary to adjust several connection weights in the learning rule. There are several learning procedures to adjust the connection weights; nevertheless, the Levenberg - Marquardt learning rule is used [14]. The rule is a variant of the Gauss-Newton optimization method. The connection weights are calculated as follows:

$$W \leftarrow W - (J^T J + \lambda I)^{-1} J\varepsilon \tag{3}$$

Where ε is the difference between the actual and required response. The Jakobian matrix for a single neuron can be written as follows:

$$J = \begin{bmatrix} \frac{\partial \varepsilon_1}{\partial w_1} & \Lambda & \frac{\partial \varepsilon_1}{\partial w_n} & \frac{\partial \varepsilon_1}{\partial w_0} \\ \mathrm{M} & & \mathrm{M} & \mathrm{M} \\ \frac{\partial \varepsilon_p}{\partial w_1} & \Lambda & \frac{\partial \varepsilon_p}{\partial w_n} & \frac{\partial \varepsilon_p}{\partial w_0} \end{bmatrix} \tag{4}$$

Parameter λ is modified based on the development of the error signal and p represents the available patterns [15, 16].

3 Experimental Procedure

A control mixture was prepared using Type I Portland cement, a limestone coarse or fine aggregate (with a particle size of 6.35 mm in the first case, and of 4.76 mm in the second case), and water. The particle size analysis of all materials used was carried out according to the Mexican NMX-C-077-ONNCCE standard. The four mix-

tures were prepared by changing the weight fraction of Portland cement. It must be stated that a water/cement ratio equal to 0.8 was used in all mixtures.

In this study, a quick and simple conventional slab-making process was employed. In this process, each one of the studied mixtures was poured into a suitable mold, followed by vibration and compacting under manual pressure applied during a fixed period of time. In this way, concrete slabs with dimensions 62x25x14cm were obtained. After extracting the formed slabs from the mold, these were water cured for time periods of up to 28 days in order to promote the hydration of Portland cement.

The flexural strength tests were conducted according to the ASTM C78 Standard [17]. A Tinus Olsen Machine was used to conduct these tests. The flexural strength of the slabs with finished type tile was measured after a period of water curing of 3, 7, 14 or 28 days. Three tests were carried out for each curing time, per each mixture composition.

The neural network architecture has two inputs and one output (Fig. 1). The first input represents the percent of cement used to build slabs. The second input represents the curing day. The output represents the flexion supported. The experimental data have samples of percentages, curing days (only 3, 7, 14 and 28 curing days) and the resultant flexion. From the experimental data 48 patrons were extracted in order to train the neural network and 16 patrons were used as validation data.

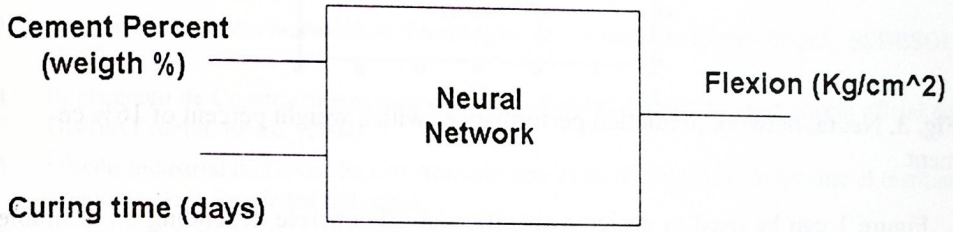


Fig. 1. Architecture of the neural network to approximate the flexion strength.

4 Results and Discussion

A Matlab® neural networks toolbox was used to generate a structure of two layers with 50 inner neurons with tangential activation function and Leverberg-Maquart learning rule. The neural network was used to predict the flexion strength of some mixture not used in training the procedure. Figure 2 shows the prediction of flexion strength through 38 days using an 11% cement mixture. Figure 3 shows the ´prediction of flexion strength through 28 days using 16 % of cement.

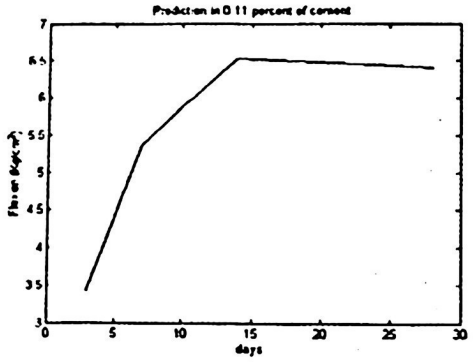


Fig. 2. Neural network prediction performances with a weight percent of 11% cement.

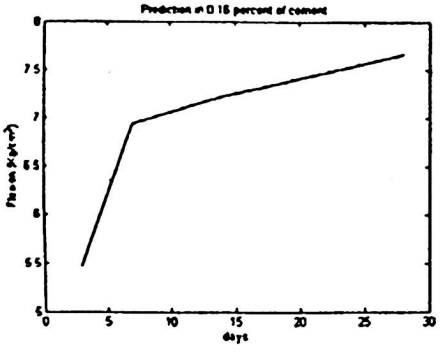


Fig. 3. Neural network prediction performances with a weight percent of 16% cement.

Figure 1 can be used to design a specific slab of concrete depending on desirable flexion strength characteristics. The plot was generated by setting the curing day input to be 28 days and generating an input set of cement percents in a range between 0.1% to 0.2%.

5 Conclusion

The generation of test slabs takes time and represents a high cost. The neural network can be used to predict the flexion strength in percent of cement not used in the experiment saving cost and time to design mixtures to build slabs of high performance in flexion strength and fulfill the requirements of the construction norms. The plot can facilitate this design. The use and comparison with other neural networks and regression models is a future work.

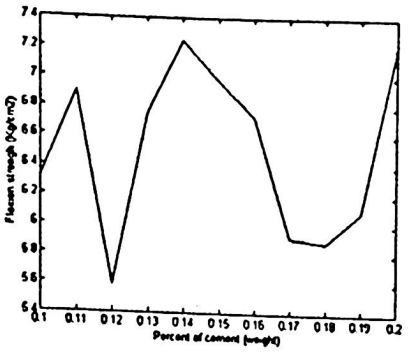


Fig. 4. Different percent of cement versus the flexion strength predicted.

References

1. X Censo General de Población y Vivienda, http://www.inegi.gob.mx/, (2000).
2. Corzo, Vivienda Campesina en México, (1978).
3. Catálogo I Concurso Nacional de Tecnologías de Vivienda de Interés Social, SEDESOL, (1992)
4. Reglamento de Construcciones para el Distrito Federal. Publicado en la gaceta oficial del Distrito Federal, Enero. (2004)
5. Diseño Industrial de Bovedilla con Acabado tipo Teja. N° Registro 15668 ante el Instituto Mexicano de la Propiedad Industrial.
6. J Arrieta F, E Peñaherrera D. Fabricación de Bloques de Concreto con una mesa vibradora, PC-CISMID, (2001)
7. G.B. Barrera, G. Carreón, A. Ruíz, "Control de calidad de piezas cerámicas por medio de análisis de vibraciones y redes neuronales artificiales", Computación y Sistemas, Vol 8, No. 3, pp187-197, (2005)
8. Hewayde, E, Author Hewayde E. Hewayde, E. , Nehdi, M, et al. Neural network prediction of concrete degradation by sulphuric acid attack. STRUCT INFRASTRUCT E 3 (1) MAR (2007) 17-27.
9. Compressive Strength Prediction of Building Blocks from Lightweight Raw Materials: A Neural Network Approach Jorge L. Acevedo D., COMIMSA, FIME-UANL, Mexico Luis M Torres-T., COMIMSA, FIME-UANL, Mexico Lauren Y Gómez Z., COMIMSA, FIME-UANL, Mexico Electronics, Robotics and Automotive Mechanics Conference (CERMA'06) Vol. 2, (2006) 184-190
10. A. Cheang-Martínez, J.L. Acevedo-Davila, L. Treviño-Torres, F.A. Reyes Valdes. Corrosion prediction and annual maintenance optimization of concrete structural components using neural network modeling. In 1st International Conference on Advanced Construction Materials. December (2006) 3 - 6
11. Jorge Acevedo, Luis Torres-T, Mario Trejo, Cesar Juárez. Compression strength prediction of mixtures concrete with scrap tire with neural network approach. In 1st International Conference on Advanced Construction Materials. 3 - 6 December 2006.

12. J.L. Acevedo, Luis Torres, Mario Trejo Aguirre. Aplicación de redes neuronales para predecir la resistencia a la compresión de elementos constructivos: Caso bloques. In 4to. CONGRESO NACIONAL DE METALURGIA Y MATERIALES 27 AL 29 DE SEPTIEMBRE DE 2006, MONCLOVA, COAHUILA, pag. 381-388, 2006
13. Jyh-Shing Roger Jang & Chuen-Tsai Sun & Eiji Mizutani
14. Neuro-Fuzzy and Soft Computing
15. Prentice Hall, (1997)
16. Simon Haykin
17. Neural Networks: A comprehensive foundation
18. Prentice Hall, (1999)
19. Hagan, M.T., and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, Vol. 5, No. 6, 1999, pp. 989-993, 1994.
20. Fredric M. Ham & I. Kostanic
21. Principles of Neurocomputing for science and Engineering
22. McGraw Hill, (1997)
23. ASTM C78 Standard Test Method for Flexural Strength of Concrete (Using Simple Beam with Third-Point Loading).

Hardware/Software Partitioning of Coarse-Grained Reconfigurable System Using Evolutionary Ant Colony Optimization

Dawei Wang, Yafei Cao, Sikun Li

College of Computer Science, University of Defence Technology,
410073 Changsha, P.R.China
daweiwang@nudt.edu.cn, yfcao@nudt.edu.cn, lisikun@263.net.cn

Abstract. We present a hardware/software partitioning approach for coarse-grained reconfigurable SoC design using an evolutionary Ant Colony Optimization with Genetic Algorithm, namely eACOGA. The architecture of reconfigurable SoC mainly consists of micro-processor, dynamic reconfigurable process element array and memories. We build an automatic configuration and optimization framework for coarse-grained reconfigurable SoC design, the kernel of which is eACOGA algorithm. eACOGA uses genetic algorithm to evolve the parameters of ant colony optimization, and makes use of such advantages as positive feedback and efficient convergence of ant colony optimization to search for optimal partitioning solutions. It utilizes the advantages of both ant colony optimization and genetic algorithm, which avoids dropping into local optimization solutions. Experiments show eACOGA improves the quality and efficiency of coarse-grained reconfigurable SoC hardware/software partitioning.

1 Introduction

Due to its potential to greatly accelerate a wide variety of applications, reconfigurable computing has become a subject of a great deal of research [1] [2]. Coarse-grained reconfigurable architectures can achieve more speedup and power saving than fine-grained reconfigurable architectures, while cost more design efforts. With more and more coarse-grained reconfigurable architectures have been proposed [3-5], the new SoC architectures patterns like "RISC + reconfigurable arrays" bring new challenges on existing SoC design methodology and tools.

Hardware/software partitioning is the key problem of coarse-grained reconfigurable SoC design. Its main task is to map tasks to hardware/software architectures for the high performance of SoC System. Recently, most research efforts use heuristic algorithms for hardware/software partitioning. These algorithms use heuristic information to guide the search process for converging to the optimal solutions step by step, so that they have the capability of handling the delay/area/power consumption trade-off.

In this paper we describe a new hardware/software partitioning approach for coarse-grained reconfigurable SoC using ant colony optimization. We have developed a framework for the automatic configuration and optimization of hardware/software

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 45–55

Received 13/06/07
Accepted 31/08/07
Final version 20/09/07

partitioning. A key component of this framework is an evolutionary ant colony optimization algorithm for hardware/software partitioning. Experimental results show that our algorithm is effective in finding close to optimal solutions.

The rest of the paper is organized into 6 sections. Section 2 overviews the previous work. Section 3 shortly introduces problem definition for coarse-grained reconfigurable SoC partitioning. In section 4 we present the ant colony optimization algorithm for coarse-grained reconfigurable SoC and integrate it into an automatic reconfigurable partitioning flow. In section 5, we apply our algorithm to some typical SoC applications and give the obtained results. Finally, in section 6 we present the conclusions.

2 Related Work

There are a great number of approaches to HW/SW co-design of SoC systems, which use different techniques for partitioning. A hybrid genetic algorithm is presented for constrained hardware-software partitioning [6]. Given a high-level program and an area constraint, it considers different granularities to discover the most interesting blocks to be implemented in ad hoc functional units. G.Wang uses ant colony optimization algorithm, in which intelligent ants can cooperatively search for the global best optimal solutions using both heuristic information and pheromone information [7].

There are many approaches to HW/SW partitioning of fine-grained reconfigurable SoC. A new automated compile-time partitioning and scheduling algorithm is proposed in [8], in which three meta-heuristic algorithms, namely genetic algorithm, simulated annealing, and tabu search, are compared. Simple heuristics are presented to partition the real-time applications at task level for scheduling between CPU and FPGA so that they meet the deadline constraints for a given total system power [9]. This fast partitioning mechanism would enable dynamic reconfiguration of systems for power and performance tuning.

However, there are not many research efforts on the partitioning of coarse-grained reconfigurable SoC systems. A hyper graph covering algorithm is proposed to translate high level C program into hyper graph and use the clustering and allocation algorithm for mapping high level application on coarse-grained reconfigurable architecture [10]. While our approach makes use of ACO (Ant Colony Optimization), which is a cooperative heuristic searching algorithm by the observation on the behavior of ants, and we evolve the parameters of ACO with GA (Genetic Algorithm) to avoid the stagnancy of search.

3 Hw/Sw Partitioning of Coarse-grained Reconfigurable SOC

3.1 Problem Formulation for Reconfigurable SoC Partitioning

Usually, we use TG (Task Graph) to describe the behaviors of SoC system, rAG (reconfigurable Architecture Graph) to describe the architectures of reconfigurable SoC

system, and the mapping from TG to rAG to describe hardware/software partitioning of reconfigurable SoC [11].

Definition 1 (TG) A task graph $TG = (T, E, C, P)$ consists of a set of nodes T, a set of directed edges $E \subseteq (T \times T)$, the configuration of nodes C, and a set of node ports P.

The nodes T represents the tasks of SoC system, and the edges E represents control/data flow of tasks' communication.

Definition 2 (rAG) A reconfigurable architecture graph $rAG = (C^{I}, C^{I/E}, C^{M}, RN)$ consists of microprocessor computation resources C^{I}, reconfigurable computation resources $C^{I/E}$, memories C^{M}, and route networks RN.

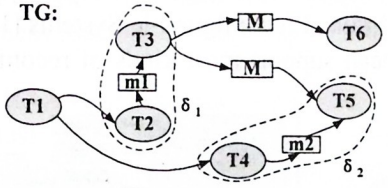
Definition 3 (*k* way partition) Given a set of modules $M = \{m_1, m_2, \ldots, m_n\}$, a k way partition problem is to find a set of clusters $P = \{p_1, p_2, \ldots, p_k\}$, which meets:

$$\begin{cases} p_i \subseteq M, & 1 \le i \le k \\ \bigcup_{i=1}^{k} p_i = M & \\ p_i \bigcap p_j = \Phi, & 1 \le i, j \le k, i \ne j \end{cases} \tag{1}$$

In the problem of reconfigurable SoC hardware/software partitioning, M presents TG, and P presents rAG. Generally, the performance objects of hardware/software partitioning are various due to the complexity of application field. In this paper, we assume the performance object is to minimize the weighted cost sum of running time and area of all the tasks. In this way, we define the objective function as follows:

Objective Cost Function $= \sum w_t * time(i) + w_a * area(i)$:

In above formulation, w_t presents weight for time, w_a presents weight for area, time(i) presents time cost of task i, and area(i) presents area cost of task i. A typical partitioning for coarse-grained reconfigurable SoC is shown in figure 1.



Mapping:

V	C	V	C
T1	Processor	T5	PEs
T2	PEs	T6	Processor
T3	PEs	m1	LM
T4	PEs	m2	LM

$\Delta = \{\delta_1, \delta_2\}$
$\Phi' = \{(T2, \delta_1), (T3, \delta_1), (T4, \delta_2), (T5, \delta_2)\}$
$\Psi' = \{(\delta_1, PEs), (\delta_2, PEs)\}$

Fig. 1. Problem formulation for reconfigurable SoC partition

We use DAG (Directed Acyclic Graph) based bi-coloring model to describe the problem of hardware/software partitioning. Figure 2 shows that tasks mapping to microprocessor are rendered by color c1, and tasks mapping to PEs are rendered by color c2. The work of ants is to find the best coloring of all tasks that minimizing the cost of partitioning. eACOGA algorithm uses dynamic combination of ant colony optimization and genetic algorithm to optimize the search for the best coloring.

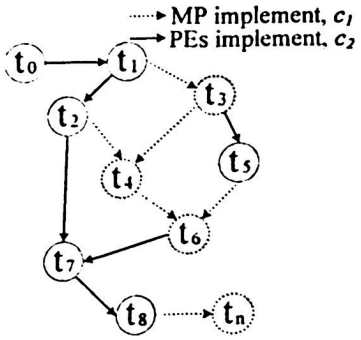


Fig. 2. Direct acyclic graph based bi-coloring model. The DAG consists of tasks nodes and their edges. All the edges are colored by two color, c_1 and c_2, which indicates mapping the tasks onto microprocessor (MP) or PEs.

3.2 Target Architecture for Reconfigurable Partitioning

We deal with reconfigurable SoC hardware/software partition by two main steps: mapping tasks to microprocessor or reconfigurable array, and partial partitioning for the tasks mapping on reconfigurable arrays.

To meet the needs of various applications, many reconfigurable architectures have been proposed, which have different details of design implementation, such as the topology of reconfigurable array, communication protocol, and the strategy of reconfiguration. To facilitate the research of reconfigurable partition problem, we abstract reconfigurable architecture and take it as the target architecture for partitioning. We design a RAAM (Reconfigurable Architecture Abstract Model) for the hardware/software partitioning of coarse-grained reconfigurable Systems [12]. RAAM has many configurable parameters, so it can support most kinds of reconfigurable architectures.

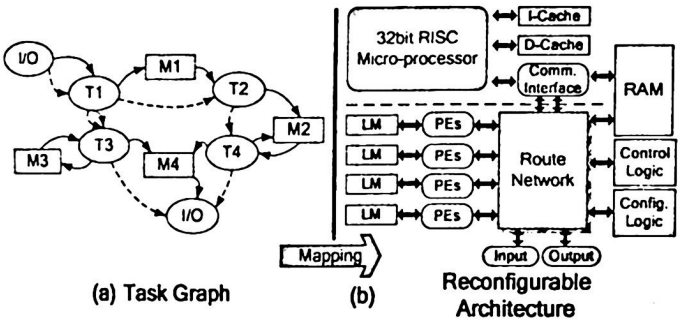


Fig. 3. Reconfigurable architecture abstract model for hardware/software partitioning. (a) task graph describes the function and behavior of reconfigurable system; (b) reconfigurable architecture consists mainly of microprocessor and coarse-grained reconfigurable PEs array.

Figure 3(a) shows the task graph that describes the function and behavior of reconfigurable systems. There are three types of task nodes: computing task, storage seg-

ment and communication task. The task nodes are connected by dependencies or channels. Figure 3(b) shows the reconfigurable architecture that consists of microprocessor and reconfigurable array. The microprocessor uses 32-bit RISC instruction set and some enhanced instructions for special propose processing. Reconfigurable array consists of PEs (Process Elements), route network, LM (Local Memory), configurable logic and control logic, etc. PEs array based reconfigurable architectures are popular because of their advantage of high performance, low power and good flexibility for embedded system applications.

4 Ant Colony Optimization For Coarse-Grained Reconfigurable SoC Partitioning

4.1 eACOGA Algorithm

ACO algorithm can find better solutions of partitioning more effectively [13]. But the strategy of random selection in constructing solutions lead to slow convergence speed. Furthermore, the principle of positive feedback can not only strengthen the solutions with better performance, but also bring on the stagnancy of search. The causation is that the main configurable parameters of the algorithm, such as α, β, ρ, Q, are set to fixed value when initializing, and it has no adaptability to various applications.

Based on existing basic ACO algorithm ([7]), we present an eACOGA approach of hardware/software partitioning for reconfigurable SoC. GA can evolve the configuration parameters of ACO algorithm by cross operation and variation operation ([9]). So eACOGA can evolve and optimize itself to search global optimal solutions.

We define the rules of eACOGA as follows:

Objective Function and Fitness Function: We define objective function as S_{best} = arg min C_p, fitness function as Fitness (p) = $1/C_p$. Where, C_p figures the cost of partition p.

Configure RAAM: According to the need of specific applications, design experts configure the task graph and reconfigurable architecture. In this step, the parameters of the tasks and architectures should be decided.

Strategy of Render to DAG: For any nodes except t_n, ants try to render the color of t_j, the subsequence of t_i. Ants achieve the work according to the global heuristic information ($\tau_{ij}(k)$) of edge e_{ij} and the local heuristic information ($\eta_j(k)$) of node t_j. The ants on node t_i will render the color of node t_j as c_k at the probability of:

$$p_{ij}(k) = \frac{\tau_{ij}(k)^{\alpha} \eta_j(k)^{\beta}}{\sum_{l=1,2} \tau_{ij}(l)^{\alpha} \eta_j(l)^{\beta}} \tag{2}$$

Where, $\tau_{ij}(k)$ is the pheromone on edge e_{ij}, α and β is the factor of them and $\eta_j(k)$ is defined as follows:

$$\eta_j(k) = 1/((w_t * time_t(k)) + (w_a * area_j(k))) \tag{3}$$

Use Genetic Algorithm to Evolve Parameters: We use genetic algorithm to evolve the parameters of ant colony optimization, such as α, β, ρ, Q. First, Configure the probability factor of cross and variation operation according to the size of population and the generation of evolution. Then by taking α, β, ρ, Q as the variable of fitness function, the best optimal partition cost as fitness function and the course of ACO as the individual, we optimize α, β, ρ, Q repeatedly until finding the best optimal solutions.

Pheromone Setting and Refreshing: We adopt MMAS (Max-Min Ant System) introduced by Thomas Stuzle ([14]), the refreshing equation of pheromone is:

$$\tau_{ij}(k)=\begin{cases}(1-\rho)*\tau_{ij}(k)+\Delta\tau_{ij}(k)_{best}\\ \tau_{ij}(k)_{max}, \text{if } \tau_{ij}(k)>\tau_{ij}(k)_{max}\\ \tau_{ij}(k)_{min}, \text{if } \tau_{ij}(k)<\tau_{ij}(k)_{min}\end{cases} \quad (4)$$

Where, ρ is the evaporation ratio of pheromone, $\tau_{ij}(k)_{max}$ ($\tau_{ij}(k)_{min}$) is maximum (minimum) strength of c_k pheromone on edge e_{ij}, and $\Delta\tau_{ij}(k)_{best}$ is increment of c_k pheromone on edge e_{ij} done by the "best ant" in current ant system algorithm iteration. According to Ant-Cycle Model, $\Delta\tau_{ij}(k)_{best}$ is defined as:

$$\Delta\tau_{ij}(k)_{best}=\begin{cases}Q/C_{p_{best}}, \text{in } p_{best}\ e_{ij} \textit{ is rendered by } c_k\\ 0, \qquad \text{otherwise}\end{cases} \quad (5)$$

```
//Input: the configuration of RAAM and DAGs .
//Outputs: the best optimal solutions for partition
//Q, α, β, ρ is the main parameters of ACO
1  Main(){
2    Generation = 0;
3    Initialize();
4    Evaluate();
5    Keep_the_Best();
6    foreach generation
7    {
8      select();
9      crossover();
10     mutate();
11     report();
12     Evaluate();
13     elitist();
14   }
15   Output the best fitness values.
16 }

17 Evaluate(){
18 for (mem = 0; mem < POPSIZE; mem++)
19 {
20   for (i = 0; i < NVARS; i++)
21     x[i+1] = population[mem].gene[i];
22   Q = x[1]; α = x[2]; β = x[4]; ρ = x[4];
23   ACO partition = new ACO;
24   partition GetAnt();
25   partition StartSearch();
26   population[mem].fitness = CostFunction_to_Fitness;
27 }

28 ACO::GetAnt(){
30   Randomly put ant into DAGs;
31   for (i = 0; i < nAntCount; i++)
32   {
33     task = rnd( nTaskCount);
34     ants[i].AddTaskIntoTabu(task);
35   }
36 }

37 ACO::StartSearch(){
38   foreach ant
39   {
40     Select_NextNode_Accordingto_heuristic_Information();
41     Moveto_NextNode();
42     Update_Tabu_Table();
43     find out the best solution of the step and put it into temp;
44   }
45   Update_Trail();
46   Find_theBest_Solutions_Of_Partition();
47 }
```

Fig. 4. Pseudo-code for eACOGA algorithm

The pseudo-code for eACOGA algorithm is shown in Figure 4. First, the configurations of RAAM and DAGs are input, and after the execution of eACOGA algorithm the best optimal solutions for partition are output. In eACOGA, ACO is built as a class, which has two main functions: GetAnt() and StartSearch(), as shown in Line 23-25. GA randomly encodes the variables of α, β, ρ, Q, as shown in Line 18-22. GA evolves the variables continuously by the operation of select, crossover and mutate, as shown in Line 6-14. In ACO, we put the ants randomly into DAGs and begin the search for best optimal solutions, as shown in the function on Line 28-36 and Line 37-47. After evaluating the fitness function values we output the best optimal solutions, as shown on Line 12 and Line 15.

4.2 Automatic Partitioning Flow

We have designed an automatic partitioning flow for mapping applications on reconfigurable SoC, as shown in Figure 5.

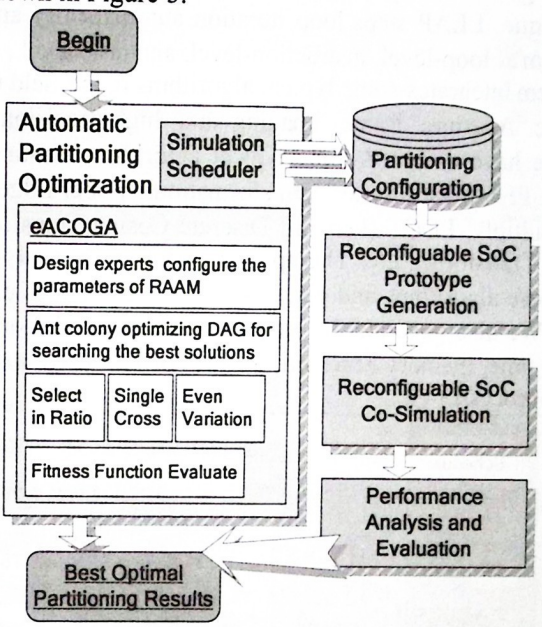


Fig. 5. Automatic partitioning flow of reconfigurable SoC. It integrates eACOGA for partitioning configuration.

First, design experts configure the tasks and reconfigurable architecture of RAAM. Then, application specific reconfigurable SoC prototype is generated according to existing reconfigurable architecture templates. Finally, we run reconfigurable SoC transaction level co-simulation and output the best optimal partitioning solutions.

The automatic partitioning flow has two main advantages:

(1).For each individual of genetic population in eACOGA, the flow of partition and reconfigurable SoC co-simulation can run automatically. When some constraints cannot be met, experts can request to stop the simulation.

(2).Transaction level simulation in SystemC can describe various behaviors of reconfigurable SoC with faster speed and nicer accuracy. Architecture template enhances reuse of existing SoC design and achieves exploration speedup well.

5 Experimental Results

5.1 Target Architecture and Benchmarks

We use eACOGA algorithm for the partitioning of a reconfigurable SoC system, which consists mainly of 32-bit RISC microprocessor called Estar and reconfigurable arrays called LEAP, as shown in Figure 7 [15]. Both Estar.and LEAP are developed by our research group. We use Estar for common computing. It has 8KB instruction cache and 8KB data cache, 266M Hz and 220mW of CPU core. Besides, we use LEAP for reconfigurable computing. It can accelerate applications through loop self-pipelining technique. LEAP steps loop iteration automatically and has the ability to exploit parallelism at loop-level, instruction-level, and task-level.

The SoC system integrates some typical algorithms in the field of Software Defined Radio, Synthetic Aperture Radar imaging and high-precision digital image encode/decode. We have designed some typical algorithms running on reconfigurable systems, such as FFT (Fast Fourier Transformation), Sobel Edge Detection, Median Filter, Matrix Multiply, FDCT (Forward Discrete Cosine Transform), IDCT (Inverse Discrete Cosine Transform), etc. The application program analysis and pick up tool can recognize these algorithms and generate them as the nodes of DAG, such as T1, T2 in Figure 6. We also generate the attributes of these nodes, such as configuration time, computing time, memory accessed, number of PEs used, and execution time on embedded microprocessor etc.

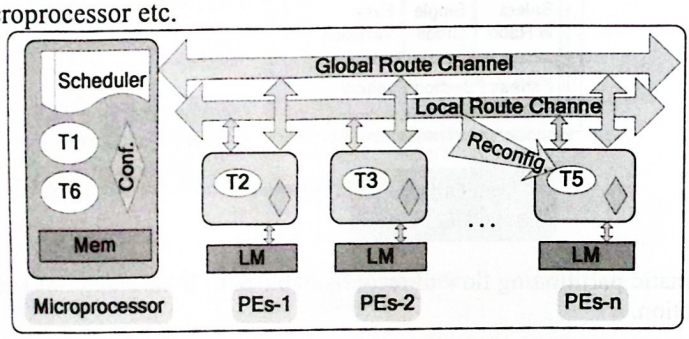


Fig. 6. A typical partitioning result of reconfigurable SoC. The task of T1 and T6 are mapped onto microprocessor, and T2, T3, and T5 are mapped onto PEs.

We have tested the performance of these typical algorithms on Estar and LEAP and translate execution time into time cost and resources used into area cost. In eACOGA algorithm, we set $w_t = 1$, $w_a = 10$, ACO iteration counts = 100, GA population size = 5, GA max generation = 50, GA cross probability factor = 0.8, and GA variation probability factor = 0.15.

Table 1. The performance of typical algorithms. Execution time of them on Estar and LEAP are shown. cPE and mPE discribes the PE for computation and the PE for memory.

Typical Algorithms	Execution Time		Resources (cPE, mPE)
	Estar (Mcycle)	*LEAP (Kcycle)*	
512 point FFT	32.320	6.721	10c4m
1024 point FFT	72.353	12.802	10c4m
Edge Detection (320x240)	39.720	216.958	16c7m
Edge Detection (480x360)	87.898	474.205	16c7m
Median Filter (320x240)	1580.368	220.010	30c7m
Median Filter (480x360)	3590.500	478.792	30c7m
Matrix Multiply (64x64)	54.315	79.141	30c10m
Matrix Multiply (128x128)	2522.258	318.901	30c10m
FDCT	2433.389	2838.905	30c10m
IDCT	2437.417	2839.044	30c10m

5.2 Result Analysis

We generate some test DAGs, the nodes of which consist of typical algorithms in Table 1. We set the parameters region of α, β, ρ, Q respectively as [5, 1], [5, 1], [0.8, 0.2], [100, 40].

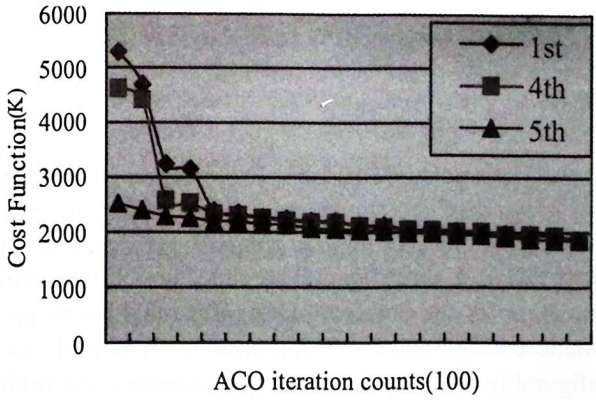


Fig. 7. The evolution curve of eACOGA for partition. The convergence speed of 5th ACO is much faster than that of 1st and 4th.

We achieve the object of hardware/software partitioning using eACOGA, as shown in Figure 7. The total number of tasks is 10, and ant counts = 6. The global best optimal solutions are 1862843, in which Q = 70.540, α = 4.836, β = 3.580, ρ = 0.738. The average value of 1st is 1918421, by evolution the average value of 4th is 1891026, and that of 5th is 1890922.

To compare the quality of eACOGA with that of other researches, we select ACO algorithm in literature [8]. We set ACO parameters as: Q = 1000, α = 1, β = 1 and ρ = 0.8. Figure 8 show that eACOGA has better ability than ACO in searching for the global best optimal solutions. The algorithm of ACO gets into local best optimal solutions (1905396). However, eACOGA can find the global best optimal solutions (1862843) effectively for the advantage of self-adaptive optimization. Besides, another algorithm we have researched, called initACO (ACO with init pheromone), has the performance between them [16].

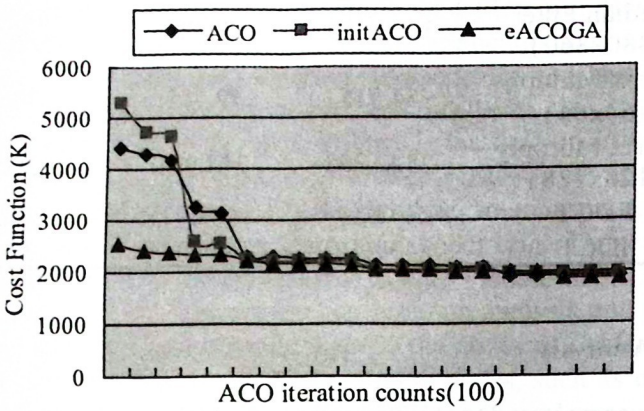


Fig. 8. Comparing eACOGA with ACO and initACO. It shows eACOGA can find optimal solutions with fewer iteration counts.

6 Conclusions

ACO has been well proved to be suitable for fine-grained reconfigurable SoC partitioning, but not mentioned for coarse-grained reconfigurable SoC. This paper proposes a hardware/software partitioning approach of coarse-grained reconfigurable system using eACOGA. A reconfigurable architecture abstract model is proposed as the target architecture for the problem of reconfigurable hw/sw partitioning. We also build an automatic configuration and optimization framework for mapping applications on reconfigurable SoC. By configuring parameters and running simulation we can get global best optimal partitioning solutions. It can save the time of waiting for manual simulation.

The algorithm of eACOGA can evolve the main control parameters (α, β, ρ, Q) of ACO, so that it can find global best optimal solutions efficiently and rapidly. It overcomes the disadvantage of ACO that be inclined to get into local best optimized solu-

tions. Furthermore, the method combining ACO and GA that yields even better results than using each of the algorithms individually.

References

1. Compton Katherine,Hauck Scott. Reconfigurable Computing: A Survey of Systems and Software. *ACM Computing Surveys*, 2002, 34(2):171-210.
2. R. Hartenstein. "A Decade of Reconfigurable Computing: a. Visionary Retrospective," In *Int'l Conf. on Design, Automa-. tion and Test in Europe (DATE'01)*, Munich, Germany,. March 12-15, 2001, 642-649.
3. Volker Baumgarten, G. Ehlers, F. May, Armin Nückel, Martin Vorbach, and Markus Weinhardt: PACT XPP - A Self-Reconfigurable Data Processing Architecture. *The Journal of Supercomputing*, 2003, 26(2): 167-184.
4. H Singh, M Lee, G Lu, F J Kurdahi, N Bagherzadeh, E Filho, R Maestre. Morhposys: case study of a reconfigurable computing system targeting multimedia applications. *Proc. Design Automation Conference (DAC'00)*, Los Angeles, California, 2000, 573-578.
5. J.R. Hauser, J. Wawrzynek. Garp: a MIPS processor with a reconfigurable coprocessor. *5th IEEE Symposium on FPGA-Based Custom Computing Machines (FCCM'97)*, 1997, 12-21.
6. Pierre-Andre Mudry, Guillaume Zufferey, Gianluca Tempesti: A Hybrid Genetic Algorithm for Constrained Hardware-Software Partitioning. *Proceedings of the IEEE Workshop on Design and Diagnostics of Electronic. Circuits and Systems*, 2006, 3-8.
7. Gang Wang, Wenrui Gong and Ryan Kastner, "System Level Partitioning for Programmable Platforms Using the Ant Colony Optimization", In *13th International Workshop on Logic and Synthesis (IWLS'04)*, 2004.
8. Theerayod Wiangtong. *Hardware/Software Partitioning and Scheduling for Reconfigurable Systems*. Ph.D. thesis, University of London, UK, 2004.
9. Pramod K., S. Acharya, R. N. Mahapatra, "A Partitioning Algorithm for Power constrained Reconfigurable Real-Time Systems", *Journal of Microprocessor and Microsystems*, 2005.
10. Yuanqing Guo, Gerard J. M. Smit, Hajo Broersma, Paul M. Heysters: A graph covering algorithm for a coarse grain reconfigurable system. *LCTES 2003*, 199-208.
11. Michael H. Eisenring. *Communication channel synthesis for heterogeneous embedded systems*. Ph.D. thesis, SWISS Federal institute of Technology ZURICH, No.14640, 2002.
12. M. Kaul, V. Srinivasan, etal, "Partitioning and Synthesis for Run-Time Reconfigurable Computers Using the SPARCS System", In *Proceedings of the 1998 Military and Aerospace Applications of Programmable Devices and Technologies Conference (MAPLD'98)*, NASA Goddard Space Flight Center, Sept 15-16, 1998.
13. M Dorigo, V Maniezzo, and A Colorni. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans on Systems, Man and Cybernetics, Part-B*. 1996, 26(1), 29~41.
14. T Stutzle, H H Hoos, et al. MAX-MIN Ant System. *Future Generation Computer System*. 2000, 16(8): 889~91.
15. Yong Dou, Xicheng Lu, "LEAP: A Data Driven Loop Engine on Array Processor", *The 4th Int'l Conf on Parallel and Distributed Computing, Applications and Technologies (PDCAT'03)*, 2003.
16. Xiong Zhihui , Li Sikun , Chen Jihua, "Hardware/Software Partitioning Based on Ant Optimization with Initial Pheromone", Journal of Computer Research and Development, 2005, 42(12):2176-2183.

Discovering Efficient Turbo Code Interleavers by Genetic Algorithms

Pavel Krömer[1], Václav Snášel[1], Nabil Ouddane[2]

[1]Department of Computer Science, Faculty of Electrical Engineering and Computer Science
VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic
{pavel.kromer.fei, vaclav.snasel}@vsb.cz
[2]Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic
nabil.ouddane.st1@vsb.cz

Abstract. Since the appearance in 1993, first approaching the Shannon limit, the Turbo Codes give a new direction for the channel encoding field, especially since they were adopted for multiple norms of telecommunications, such as deeper communication. To obtain an excellent performance it is necessary to design robust turbo code interleaver. We are investigating genetic algorithms as a promising optimization method to find good performing interleaver for the large frame sizes. In this paper, we present our work, compare with several previous approaches and present experimental results.

Keywords: Genetic Algorithms, Turbo Codes interleaver optimization

1 Introduction

It is known that the encoding block on the transmission scheme is one of the complex operations. Channel encoding adds redundancy symbols to the message to be transmitted causing diminution to spectral efficacy of the transmission. The Turbo Codes (TC) was a new tendency in the channel encoding field and they have become a reference soon after their introduction. Their original name comes from their first structure introduced and described in [1], namely concatenated convolutive recursive systematic codes with iterative decoding.

Turbo codes offer the best compromise between structure (concatenation) and randomness created by the interleaver. Its characteristic iterative decoding process is among the principal performance factors of the turbo codes. The significant characteristics of turbo codes are small bit error rate (BER) achieved even at low signal to noise ratio (Eb/N0) and the error floor at moderate and high values of Eb/N0. Previous studies proved that the random interleaver can be in certain cases more efficient than other channel encoding schemes [2].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 56–65

Received 16/06/07
Accepted 31/08/07
Final version 20/09/07

In this paper, genetically evolved interleaver will be compared to random interleaver by the means of BER to evaluate its efficiency. The increase of the interleaver size gives better performance and better interleaving gain while worsening latency. The relation (1) illustrates the influence on the latency

$$t_d = \frac{K_f}{R_b} N_i \tag{1}$$

where R_b is the code bit rate, K_f stands for the frame size and N_i is the number of the decoding stages. The performance of the turbo codes depends on two principal parameters, first is the code spectrum, and the second is decorrelation between the external information at the same number of iterations.

The optimization can be used for the amelioration of performance and the diminution of the matrix stature with safe performance. This second point is very interesting for multimedia real-time transmission systems over satellite because the interleaving matrix makes a considerable diminution of the codec complexity and delay. Interleaver matrix sizes vary from tens to ten-thousands of bits. It is highly inefficient to test all the possible input vectors (2^N) with all the possible interleaver matrices ($N!$), requiring $2^N.N!$ tests. Therefore, advanced interleaver optimization methods are required.

Some results based on genetic algorithm exploitation, were previously obtained in [3] but only for small value of N (N=50), achieving a performance gain of 0.1 dB. Recent researches [4] focused on larger frame sizes and it means also larger interleaver length N, which is more promising for real time communication over satellite. The choice of the objective function was based on the maximizing of the determinist performance parameter which is free distance. Such optimization is complicated due the fact that free distance calculation is non-trivial and complex task.

In following, we are using similar parameters and compare our results to interleavers discovered by the means of previous methods. In the next section is presented short overview of a turbo code system.

1.1 Turbo Code System

Figure 1a represents convolutive encoder used in the past experiences, with the dimension of the memory effect $\upsilon = 4$, constraint length $L = (\upsilon + 1) = 5$ and rate $r = \frac{k}{n} = \frac{InputSymbols}{OutputSymbols} = \frac{1}{2}$. The encoder is punctured for rate of 1/3 to get better maximal free distance.

The turbo encoder presented in Figure 1b, is the same as first turbo encoder used in [5]. It is composed of parallel concatenation of two convolutive systematic recursive codes connected with an interleaver in between. The rate of such encoder is 1/3.

The first encoder operates directly on the input sequence, denoted by c, of length N. The first component encoder has two outputs. The first output, denoted by v_0, is equal to the input sequence since the encoder is systematic. The other output is the

parity check sequence, denoted by v_1. The interleaved information sequence at the input of the second encoder is denoted by c'.

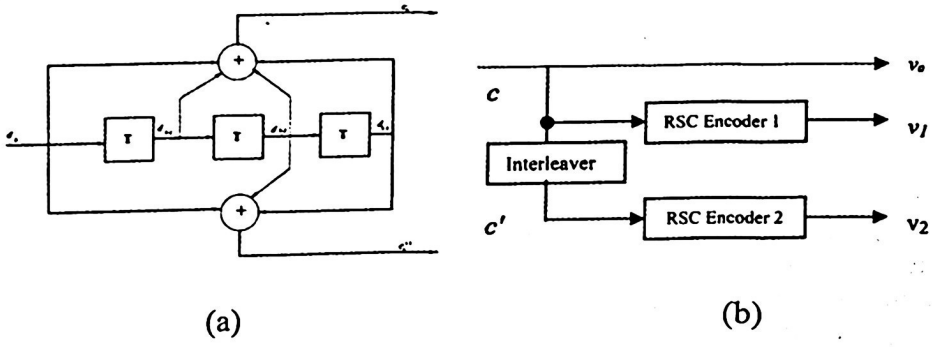


Fig. 1. Convolutive encoder (a) and turbo encoder (b)

Only the parity check sequence of the second encoder, denoted by v_2, is transmitted. The information sequence v_0 and the parity check sequences of the two encoders v_1 and v_2, are multiplexed to generate the turbo code sequence. The overall code rate is thus 1/3.

In this article will be used the Enhanced Maximum Aposteriori Probability Log Map soft decoding algorithm, a technique often used for the satellite communication decoding. More details can be found in [6].

2 Evolutionary Algorithms

Evolutionary algorithms (EA) are family of iterative stochastic search and optimization methods based on mimicking successful optimization strategies observed in nature [7, 8, 9, 10]. The essence of EAs lies in the emulation of Darwinian evolution utilizing the concepts of Mendelian inheritance for the use in computer science and applications [10]. Together with fuzzy sets, neural networks and fractals, evolutionary algorithms are among the fundamental members of the class of soft computing methods.

EA operate with population (also known as pool) of artificial individuals (referred often as items or chromosomes) encoding possible problem solutions. Encoded individuals are evaluated using objective function which assigns a fitness value to each individual. Fitness value represents the quality (ranking) of each individual as solution of given problem. Competing individuals search the problem domain towards optimal solution [8]. In the following sections will be introduced general principles common for all methods belonging to the class of evolutionary algorithms.

2.1 Evolutionary Search Process

For the purpose of EAs is necessary proper encoding, representing solutions of given problem as encoded chromosomes suitable for evolutionary search process. Finding proper encoding is non-trivial problem dependent task affecting the performance and results of evolutionary search while solving given problem. The solutions might be encoded into binary strings, real vectors or more complex, often tree-like, hierarchical structures, depending on the needs of particular application area.

The iterative phase of evolutionary search process starts with an initial population of individuals that can be generated randomly or seeded with potentially good solutions. Artificial evolution consists of iterative application of genetic operators, introducing to the algorithm evolutionary principles such as inheritance, survival of the fittest and random perturbations. Current population of problem solutions is modified with the aim to form new and hopefully better population to be used in next generation. Iterative evolution of problem solutions ends after satisfying specified termination criteria and especially the criterion of finding optimal solution. After terminating the search process, evolution winner is decoded and presented as the most optimal solution found.

2.2 Genetic Operators

Genetic operators and termination criteria are the most influential parameters of every evolutionary algorithm. All bellow presented operators have several implementations performing differently in various application areas.

Selection operator is used for selecting chromosomes from population. Through this operator, selection pressure is applied on the population of solutions with the aim to pick more promising solutions to form following generation. Selected chromosomes are usually called parents.

Crossover operator modifies the selected chromosomes from one population to the next by exchanging one or more of their subparts. Crossover is used for emulating sexual reproduction of diploid organisms with the aim to inherit and increase the good properties of parents for offspring chromosomes.

Mutation operator introduces random perturbation in chromosome structure; it is used for changing chromosomes randomly and introducing new genetic material into the population.

Besides genetic operators, termination criteria are important factor affecting the search process. Widely used termination criteria are i.e.:

- Reaching optimal solution (which is often hard to recognize)
- Processing certain number of generations
- Processing certain number of generations without improvement in population

EAs are successful general adaptable concept with good results in many areas. The class of evolutionary techniques consists of more particular algorithms having numerous variants, forged and tuned for specific problem domains. The family of evolutionary algorithms consists of genetic algorithms, genetic programming, evolutionary strategies and evolutionary programming.

2.3 Genetic Algorithms

Genetic Algorithms (GA) introduced by Holland and extended by Goldberg are wide applied and highly successful EA variant. Basic workflow of originally proposed standard generational GA is:

I. Define objective function
II. Encode initial population of possible solutions as fixed length binary strings and evaluate chromosomes in initial population using objective function
III. Create new population (evolutionary search for better solutions):
    a. Select suitable chromosomes for reproduction (parents)
    b. Apply crossover operator on parents with respect to crossover probability to produce new chromosomes (offspring)
    c. Apply mutation operator on offspring chromosomes with respect to mutation probability. Add newly constituted chromosomes to new population
    d. Until the size of new population is smaller than size of current population go back to a.
    e. Replace current population by new population
IV. Evaluate current population using objective function
V. Check termination criteria; if not satisfied go back to III.

Many variants of standard generational GA have been proposed. The differences are mostly in particular selection, crossover, mutation and replacement strategy [8].

3 Designing Genetic Algorithms for Interleaver Optimization

Genetic algorithms have been already used for interleaver matrix optimization. Durand et al. [4] used customized GA to optimize the interleaver of the size 105, comparing their results to previous interleaver design techniques. Their genetic algorithm was fully based on mutation and the crossover operator was due to complications omitted. The fitness criterion for every interleaver was maximum free distance.

Rekh et al. [3] presented recently another GA for the interleaver optimization, introducing 2-point crossover to interleaver evolution process. Nevertheless, the crossover impact is limited by necessary correction of errors created during crossover application. The fitness criterion was BER and the size of optimized interleaver 50.

In the following section will be discussed our GA for interleaver optimization and compared to previous mentioned approaches.

3.1 Interleaver GA Discussion

An interleaver of the dimension N performs a permutation of N input bits and therefore can be seen as a general permutation of N symbols. Hence, we encode interleaver for the purpose of genetic algorithm as permutation σ_N. An interleaver of the dimension N performs a permutation of N input bits and therefore can be seen as a general permutation of N symbols $\sigma_N = (i_1, i_2, \ldots, i_N)$, where $i_k \in [1, N]$ and $i_m \neq i_n$ for

all $m \neq n \in [1, N]$. The application of σ_N on input vector I_N for $N = 5$ is shown in relation (2).

$$\begin{aligned} I_5 &= (0, 1, 0, 1, 1, 1) \\ \sigma_5 &= (5, 3, 4, 1, 2) \\ O_5 &= \sigma_5(I_5) = (1, 0, 1, 0, 1) \end{aligned} \tag{2}$$

The same encoding was used also in [3]. Durand et al. in [4] do not specify their interleaver encoding, although we can conclude that they used similar interleaver representation.

We have used two types of selection: roulette wheel selection and for speeding up the convergence of the algorithm a semi-elitary hybrid selection scheme choosing one parent by elitary manners and the second by proportional manners of roulette wheel selection. Mutation is simply realized by swapping positions of two coordinates in σ_N. On the contrary, traditional crossover operators (except of uniform crossover) will corrupt the structure of permutation σ_N and hence cannot be used without some post processing used for chromosome fixing. Authors of [4] have fully omitted crossover and the crossover application in [3] lead to the need to repair every new chromosome created via crossover. This is a remarkable fact since crossover is referred as the primary operator for GA [9].

To enable the application of crossover for interleaver optimization, expecting performance increase, we have investigated the effect of uniform crossover on convergence ability of the classical interleaver optimizing GA. In the second phase, we have designed modified GA allowing the use of virtually any crossover operator for permutation evolution without corrupting the chromosomes. New crossover friendly GA is based on separation of chromosomes into groups of the same size called higher level chromosomes, (HLC's). Genetic operators are then applied on HLC's while original chromosomes act as genes as shown in Figure 2. We have tested above introduced techniques on a benchmarking problem consisting of search for an identity matrix. The results have revealed that GAs with semi-elitary selection and HLC were most efficient.

The best performing GA was used for interleaver optimization. As fitness criterion was adopted approach introduced in [3]: average BER, captured after simulated transmission of several low weight information frames.

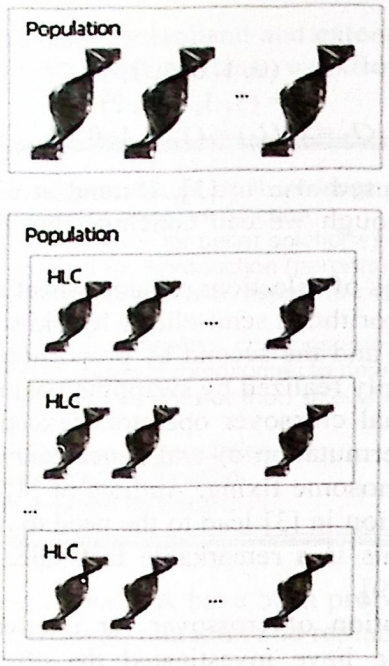


Fig. 2. Traditional population compared with HLC's

4 Experiments

An experimental framework built upon the IT++ library[1] was used to experimentally evaluate proposed interleaver generation method. We have experimented with 64, 128, 512 and 1024 bit interleavers aiming to optimize in the future as large interleaver as possible.

The settings for all optimization experiments were:

- GA with HLC and semi-elitary selection
- 100 generations
- probability of crossover 0.8
- population of 5 high level chromosomes per 6 genes
- fitness criterion was minimal BER after simulated submission of 100 random frames of weight up to 6
- the simulations were performed over additive white Gaussian noise (AWGN) channel

The AWGN channel is a good model for satellite and deep space communication links but not an appropriate model for terrestrial links. The evolved interleavers were

[1] IT++ is available at http://itpp.sourceforge.net/

evaluated by simulated transmission over AWGN channel for Eb/N0 $\in$ [0, 4] and flat Rayleigh fading channel for Eb/N0 $\in$ [0, 6]. Rayleigh fading channel was used as a reasonable model for tropospheric and ionospheric signal propagation as well as the effect of heavily built-up urban environments on radio signals [11].

In the experiments, GA with classic population and semi-elitary selection were used. Optimized interleavers were compared to random interleaver taken as reference by the means of Eb/N0 to BER ratio, captured after simulated transmission of 500 random frames.

4.1 Optimization Results

Optimization results are summarized in Figures 3 and 4. In all of them is used following notation: curve denoted as *O1* corresponds to the best interleaver found by GA with classic population, *O2* describes performance of best interleaver found using GA with HLCs and *Rand* denotes reference random interleaver. *AWGN* denoted curves illustrate experiments over additive white Gaussian noise channel and Rayleigh curves represent the experimental results measured over Rayleigh channel. In all figures can be seen that optimized interleavers perform better than reference random interleaver.

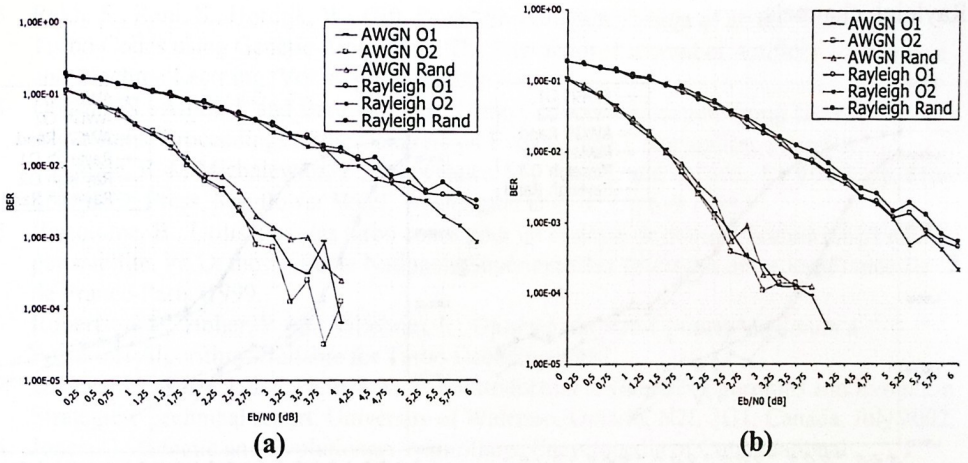


Fig. 3. 64bit (a) and 128bit (b) interleaver

5 Result Discussion

Figure 3a illustrates the binary error rate for an interleaver with the length of 64 bits; we can observe that an improvement for *AWGN* channel begin to appear from Eb/N0 = 2dB and becomes more significant for larger Eb/N0 values, especially for the interleaver obtained by second optimization method. Both optimized interleavers

overperformed the random interleaver. For BER=10^{-3} we have an Eb/N0 of approximately 3.25 dB for the random interelaver and 2.75 dB for the second optimized interleaver achieving gain of 0.5 dB. The trend is valid for Rayleigh channel experiments as well and the supremacy of interleaver *O2* is even more evident.

For 128 bits interleaver length, we can observe, as shown on Figure 3b, that for *AWGN*, the amelioration begins to be significant between the second optimization and the random interleaver from Eb/N0=2.25 dB, it means for a larger signal noise rate values. For BER=10^{-3} we have Eb/N0 equal 2.25 dB for the second optimization and 2.5 dB for the random interelaver having 0.25 dB of gain. For the Rayleigh channel transmissions, the better performance of *O1* and *O2* comparing to random interleaver becomes to be clear for greater EbN0 values (>5 dB) and *O2* is again best performing among the three.

The gain becomes more considerable for interleaver length of 512bits as shown on Figure 4a, for example in *AWGN*, we have for BER=10^{-4} the Eb/N0=2.75 dB for the second optimization method while having 3.5 dB for the random interelaver. This indicates 0.75 dB gain for 512bits length interleaver, which is for this interleaver length a remarkable result. In Rayleigh channel, the initial BER values are for all three compared interleavers almost the same while for higher EbN0, interleaver *O2* achieves permanent gain over similarly performing *O1* and random interleaver.

Similarly, optimized 1024 bit inteleavers, highlighting specially interleaver *O2*, shown at Figure 4b outperform reference random interleaver for both, *AWGN* and Rayleigh channels.

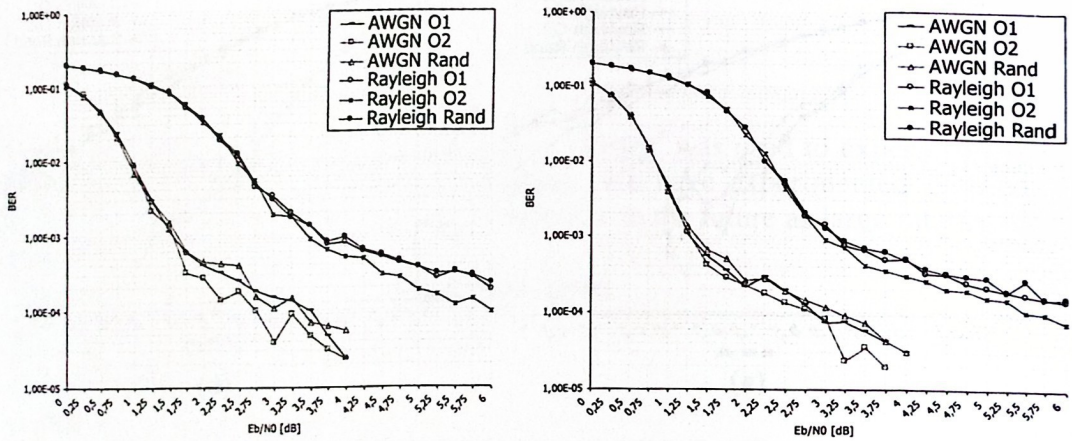


Fig. 4. 512bit (a) and 1024bit (b) interleaver

6 Conclusions

In this paper, we discussed the problem of efficient turbo code interleaver optimization by the means of genetic algorithms. Previous approaches were revised and novel modifications to existing interleaver optimizing GA improving their

convergence were introduced. Presented GA modifications are general and can be used in other application areas as well. Optimized interleavers found by introduced method were verified and compared to random interleaver by the means of BER performance. The verification utilized both, *AWGN* channel and more real-life-like Rayleigh fading channel. The optimized interleavers outperformed random interleavers having the interleaver found by presented method total winner by the means of BER to EbN0 ratio in most cases.

In the future we aim to use developed algorithm for the optimization of larger interleavers and investigate the use of minimum free distance as more competent fitness criterion. Additionally, we want to employ Rayleigh fading channel model at the optimization phase and compare obtained interleavers to interleavers evolved over *AWGN* channel. Moreover, we are investigating the general process of permutation evolution since it has numerous applications in computer science (i.e. in data compression).

References

1 Berrou, C., Glavieux, A. and Thitimajshima, P., Near Shannon limit error-correcting coding and decoding: turbo codes, Proc. Int. Conf. on Commun., pp. 1064–1070, 1993.

2 Hokfelt, J. and Maseng, T., METHODICAL INTERLEAVER DESIGN FOR TURBO CODES, International Symposium on Turbo Codes.

3 Rekh, S., Rani, S., Hordijk, W., Gift, P. and Shanmugam, Design of an Interleaver for Turbo Codes using Genetic Algorithms, The International Journal of Artificial Intelligence and Machine Learning, Vol. 6, pp. 1–5, 2006.

4 Durand, N., Alliot, J. and Bartolomé, B., Turbo Codes Optimization Using Genetic Algorithms, Proceedings of the Congress on Evolutionary Computation, edited by Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X. and Zalzala, A., Vol. 2, pp. 816–822, IEEE Press, Mayflower Hotel, Washington D.C., USA, 6-9 1999.

5 Bartolome, B., Utilisation des turbo codes pour un systeme de communications multimédia par satellite, Ph.D. thesis, Ecole Nationale Supérieure des TélécommunicationsFrance-Ile de France-Paris, 1999.

6 Robertson, P., Hoher, P. and Villebrun, E., Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding, 1997.

7 Dianati, M., Song, I. and Treiber, M., An Introduction to Genetic Algorithms and Evolution Strategies, Technical report, University of Waterloo, Ontario, N2L 3G1, Canada, July 2002.

8 Jones, G., Genetic and Evolutionary Algorithms, Encyclopedia of Computational Chemistry, edited by von Rague, P., John Wiley and Sons, 1998.

9 Mitchell, M., An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.

10 Bodenhofer, U., Genetic Algorithms: Theory and Applications, Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter 2003/2004.

11 Proakis, J. G., Digital Communications, McGraw-Hill, New York, 4th edn., 2001.

A Comparative Study of a Fuzzy Similarity Measure Aggregated with Fuzzy Implications Applied to Shape Recognition

Leila Baccour, Wafa Maghrebi, Slim Kanoun and Mohamed Adel Alimi

National School of Engineering, University of Sfax, Tunisia
leila.baccour@ieee.org, wafa.maghrebi@fsegs.rnu.tn,
slim.kanoun@enis.rnu.tn, adel.alimi@enis.rnu.tn

Abstract. The choice of a similarity measure is very important to compare between two samples. In addition, the similarity between many fuzzy sets (i.e., many linguistic variables) needs aggregation operators which can influence the fuzzy similarity results. In this paper we compare results of fuzzy implications aggregated to a fuzzy similarity measure applied to shapes recognition which are described by an extended curvature scale space (CSS) descriptor. We present experimental results on the vision Speech and Signal Processing Surrey University database.

Key words: Fuzzy similarity measure, Fuzzy implication, Shape recognition

1 Introduction

In several system of recognition, clustering, classification, etc. a similarity measure is used. The similarity of two samples is often evaluated by measuring a distance between their features. Thus, two samples are not considered similar if the difference between their sets of features is obvious.
Selecting the suitable distance or similarity is not a trivial task. Our purpose is to use fuzzy similarity measure to compare between shapes. A fuzzy similarity measures scale the equality degree between two fuzzy sets. In literature, many fuzzy similarity measures were presented and discussed e.g. [1, 2]. In this paper, we aggregate a fuzzy similarity measure with fuzzy implications to recognize shapes. We present and compare results for each aggregation.
In section 2, we present the description of the shape database which is followed by the construction of correspondent fuzzy database. In section 4, we give an overview of some fuzzy similarity measures from literature and some operations on fuzzy sets. Thus, we present a comparative study of experimental results of similarity measures aggregate with fuzzy operators.

2 Database Shape Presentation

We use the Vision, Speech and Signal Processing Surrey University database, which contains about 1100 images of marine creatures. Each image shows one

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 66–76

Received 20/07/07
Accepted 31/08/07
Final version 21/09/07

distinct species on a uniform background. Every image is processed to recover the boundary edge, which is then represented by an extended curvature scale space (CSS) descriptors.

2.1 Curvature Scale Space Descriptors

Introduced by Mokhtarian et al [3, 4], the CSS descriptors register the concavities of a curve as it goes through successive filtering. The role of filtering is to smooth out the curve and gradually eliminate concavities of increasing size. More precisely, given a form described by its normalized planar contour curve.

$$\Gamma(u) = \{x(u), y(u) | u \in [0,1]\} \ . \tag{1}$$

The curvature at any point u is defined as the tangent angle to the curve and is computed as:

$$k(u) = \frac{x_u(u)\, y_{uu}(u) - x_{uu}(u)\, y_u(u)}{\left(x_u(u)^2 + y_u(u)^2\right)^{\frac{3}{2}}} \ . \tag{2}$$

To compute its CSS descriptors, a curve is repeatedly smoothed out using a Gaussian kernel g(u,σ). The contour of the filtered curve is represented as:

$$\Gamma(u) = \{x(u,\sigma), y(u,\sigma) | u \in [0,1]\} \ . \tag{3}$$

where, x(u,σ) and y(u,σ) represent the result of convolving x(u) and y(u) with g(u,σ), respectively. The curvature k(u,σ) of the smoothed out curve is represented as:

$$k(u,\sigma) = \frac{x_u(u,\sigma)\, y_{uu}(u,\sigma) - x_{uu}(u,\sigma)\, y_u(u,\sigma)}{\left(x_u(u,\sigma)^2 + y_u(u,\sigma)^2\right)^{\frac{3}{2}}} \ . \tag{4}$$

The main idea behind CSS descriptors is to extract inflection points of a curve at different values of σ. As σ increases, the evolving shape of the curve becomes smoother and we notice a progressive disappearance of the concave parts of the shape until we end up with a completely convex form (Fig. 1.). Using the curve's multi-scale representation, we can locate the points of inflection at each scale (i.e. points where k(u,σ) = 0). A graph, called CSS image, specifying the location u of these inflection points vs. the value of σ can be created:

$$I(u,\sigma) = \{(u,\sigma) | k(u,\sigma) = 0\} \ . \tag{5}$$

Figure 2 shows the CSS image corresponding to the shape shown in Fig. 1. Different peaks present in the CSS image correspond to the major concave segments of the shape. The maxima of the peaks are extracted and used to describe the input shape. Even though the CSS descriptors have the advantage of being invariant to scale, translation, and rotation, and are shown to be robust and tolerant of noise, they are inadequate to represent the convex segments of a shape. In addition, the CSS descriptors can be considered as local features and hence do not

capture the global shape of an image contour. The following section presents a remedy (the extended CSS descriptors) for these drawbacks.

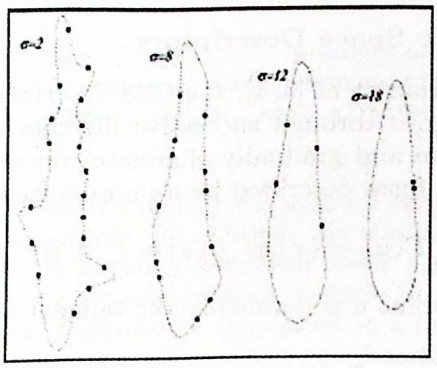


Fig. 1. Evolution of the form according to the smoothing scale σ

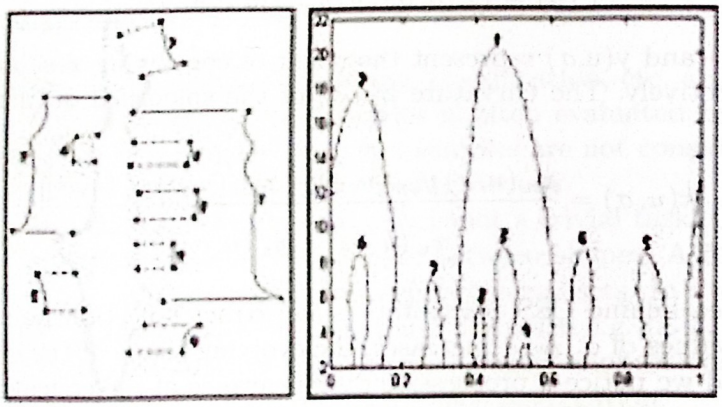


Fig. 2. Creating the CSS image of a shape

2.2 Extended CSS Descriptors

Kopf et al [5] presented a solution to remedy the inability of the CSS descriptors to represent convex segments of a shape. The idea they proposed is to create a dual shape of the input shape where all convex segments are transformed to concave segments. The dual shape is created by mirroring the input shape with respect to the circle of minimum radius R that encloses the original shape (Fig. 3.). More precisely, each point (x(u),y(u)) of the original shape is paired with a point (x'(u),y'(u)) of the dual shape such as the distance from (x(u),y(u)) to the

circle is the same from (x'(u),y'(u)) to the circle. The coordinates of the circle's centre O(M_x, M_y) are calculated as:

$$M_x = \frac{1}{N}\sum_{u=1}^{N} x(u) \quad . \tag{6}$$

$$M_y = \frac{1}{N}\sum_{u=1}^{N} y(u) \quad . \tag{7}$$

The projected point (x'(u),y'(u)) is located at

$$x'(u) = \frac{2R - D_{x(u),y(y)}}{D_{x(u),y(y)}}\left(x(u) - M_x\right) + M_x \quad . \tag{8}$$

$$y'(u) = \frac{2R - D_{x(u),y(u)}}{D_{x(u),y(u)}}\left(y(u) - M_y\right) + M_y \quad . \tag{9}$$

where, $D_{x(u),y(u)}$ is the distance between the circle's centre and the original shape pixel.

Since CSS descriptors as considered local features, we decided to use two extra global features: circularity and eccentricity. Circularity is defined as:

$$cir = \frac{p^2}{A} \quad . \tag{10}$$

where, P is the perimeter of the shape and A is its area. Eccentricity is defined as:

$$ecc = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}} \quad . \tag{11}$$

where, λ_{max} and λ_{min} are the eigenvalues of the matrix A

$$A = \begin{bmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{bmatrix} \quad . \tag{12}$$

$\mu_{2,0}$, $\mu_{1,1}$, and $\mu_{0,2}$ are the central moments of the shape defined as:

$$\mu_{p,q} = \sum_{x}\sum_{y}(x - \bar{x})^p (y - \bar{y})^q \quad . \tag{13}$$

with $\bar{x}$ and $\bar{y}$ representing the coordinates of the shape's centroid. The eccentricity feature is size, rotation and translation invariant.

The extended CSS descriptors we used in the image indexing are a combination of four sets of features:

- Circularity feature (global feature)
- Eccentricity feature (global feature)
- CSS descriptors of original shape (local features)
- CSS descriptors of dual shape (local features)

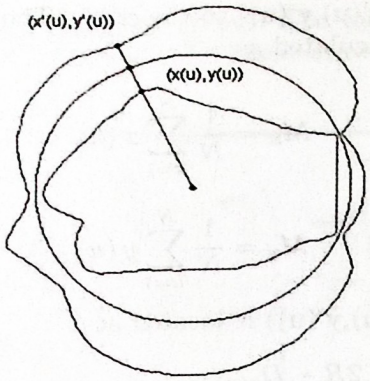


Fig. 3. Creating a dual shape with respect to an enclosing circle

3 Construction of Fuzzy Shapes Database

In this section, we are interesting to create a fuzzy shapes database. This stage permits to pass from real database constructed by real values for extended CSS descriptors to a fuzzy database. This last data are values between 0 and 1, appointing membership degrees to fuzzy sets [6]. Thus, every feature of shapes is represented with fuzzy sets which design fuzzy membership function.
To choose the membership functions, we divide database shape on two databases: the first constituted of 734 shapes serves as training database and the second constituted of 366 shapes serves as test database. Every shape is represented by the extended CSS descriptor defined in second section.
To represent fuzzy sets of concavity and convexity we think that abscises are not important to retain for matching. Because, every shape can take different positions, so two shapes can be similar if concavity ordinates are equal independently of abscises (i.e. one abscises can be on the right, other on the left). So, we retain for each shape: eccentricity, circularity, ordinate concavity points and ordinates convexity points.
We find that values of concavity, convexity and circularity can be represented by three fuzzy sets: low medium and high. However, the eccentricity is also represented by two fuzzy sets: low and high.
We choose trapezoidal function to represent fuzzy sets. We present below (Fig. 4.) the fuzzy sets for the feature convexity.

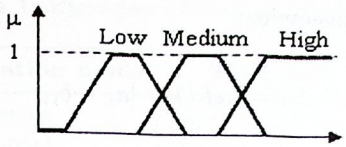


Fig. 4. Convexity membership functions

The real values of CSS descriptors of each shape are fuzzified by calculating the membership degrees to correspondent functions.

4 Overview of Some Fuzzy Similarity Measures

Many fuzzy similarity measures are presented in literature; in the sequel we present some of them.
Let $U = \{x_1, x_2, \ldots, x_n\}$ a discourse universe, A and B tow fuzzy sets in the universe of discourse defined as follow:
$A = \{(x, \mu_A(x)) | x \in U, \mu_A(x) \in [0,1]\}$, $B = \{(x, \mu_B(x)) | x \in U, \mu_B(x) \in [0,1]\}$
Suppose a and b the vector representations of the fuzzy sets A and B defined as a=$(a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, $a_i, b_i \in [0,1]$, $i = 1, 2, \ldots, n$.

- Measures proposed in [7]

$$S_1 = 1 - \sum_{i=1}^{n} \frac{|a_i - b_i|}{n} \quad . \tag{14}$$

$$S_2 = \max_i (\min(a_i, b_i)) \quad . \tag{15}$$

$$S_3 = \frac{a.b}{\max(a.a, b.b)} \quad . \tag{16}$$

- Measures proposed in [8]

$$S_4 = \frac{\sum_{i=1}^{n} \frac{min(a_i, b_i)}{max(a_i, b_i)}}{n} \quad . \tag{17}$$

with the convention $\frac{0}{0} = 1$

$$S_5 = \sum_{i=1}^{n} \frac{1 - |a_i - b_i|}{n} \quad . \tag{18}$$

- Measures proposed in [9]

fuzzy similarity of Lukasiewiez:

$$S_6 = inf\,(1 - |a_i - b_i|) \quad . \tag{19}$$

Fuzzy similarity of Gödel:

$$S_7 = inf\,(a_i \leftrightarrow_{Go} b_i) \quad . \tag{20}$$

where

$$a_i \leftrightarrow_{Go} b_i = \begin{cases} 1 \text{ if} & a_i = b_i \\ min\,(a_i, b_i) \text{ else} \end{cases}$$

Fuzzy similarity of Goguen:

$$S_8 = inf\,(a_i \leftrightarrow_{G} b_i) \quad . \tag{21}$$

where $a_i \leftrightarrow_G b_i = \begin{cases} 1 & \text{if } a_i = b_i \\ \frac{min(a_i,b_i)}{max(a_i,b_i)} & \text{else} \end{cases}$

5 Opertations on Fuzzy Sets

As for classical sets, operations are defined on fuzzy sets, such as intersection, union, complement, etc. In the literature on fuzzy sets, a large number of possible definitions are proposed to implement intersection, union and complement. The operators defined by Zadeh are:

- Intersection operator: $min(\mu_A(x), \mu_B(x))$
- Union operator: $max(\mu_A(x), \mu_B(x))$

Where A and B are fuzzy sets defined in precedent section.
General forms of intersection and union are represented by triangular norms (T-norms) and (T-conorms) and designs fuzzy implications.
A fuzzy implication is a function from $[0,1] \times [0,1]$ to [0,1] which determine the verity degree of the preposition $x \Rightarrow y$. Table 1 gives examples of implications used most often [10, 11].

Table 1. Examples of fuzzy implications

Implication name	Function
Lukasiewicz	min(1-x+y,1)
Reichenbach	1-x+xy
Goguen	$\begin{cases} min(\frac{x}{y}, 1) \text{ if } & x \neq 0 \\ 1 \; else \end{cases}$
Kleene-Dienes	max (1-x, y)
Brouwer-Gödel	$\begin{cases} 1 \text{ if } & x \leq y \\ y \; else \end{cases}$
Rescher-Gaines	$\begin{cases} 1 \text{ if } & x \leq y \\ 0 \; else \end{cases}$
Willmott	max (1-x, min(x,y))

6 Similarity Measures Based on Fuzzy Logic

Our objective is to measure the similarity of two shapes having fuzzified attributes according to fuzzy sets of features. Let shapes A and B be described by M linguistic variables v_i and for each linguistic variable v_i, linguistic values are defined L_k^i. Each linguistic value L_k^i is represented by a fuzzy set with a membership function $\mu_{L_k^i}$. The similarity of A and B is computed on two steps:

- Compute similarity of A and B according to one linguistic value L_k^i for all variables v_i, $S_{L_k^i}(A, B)$
- Compute similarity of A and B according to all lingistic values for all variables, S(A,B)

We obtain similarity measures $S_{L_k^i}(A, B)$ of A and B according to every linguistic value L_k^i for all variables v_i. Thus, we must find an aggregation or a relation between similarity degrees $S_{L_k^i}(A, B)$ to find the similarity S(A,B) of two shapes. So, we use implication operators to compute this similarity. We choose the similarity measurement S_6 defined in section 4 (equation (19)) to compute $S_{L_k^i}(A, B)$. Then, we aggregate the results to implication operators defined in section 5 and we compare results.

7 Experimental Results

We trial similarity measure S_6 aggregated to each implication and we compare results of matching two images from test database to 100 images of training set database (Fig. 5. and Fig. 6.). The images retrieved with the highest similarity rates (i.e., degree of similarity equal 0.5 or higher) shown in Fig. 5. b and Fig. 6. b obtained, using S_6 aggregated to Gödel implication and the images retrieved with the highest similarity ranks shown in Fig. 5. c and Fig. 6. c obtained, using S_6 aggregated to Willmott implication.

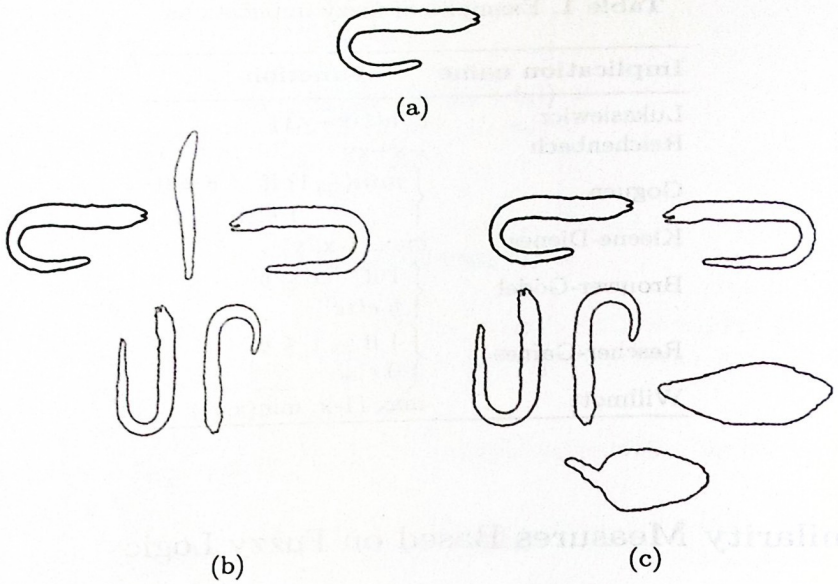


Fig. 5. (a) Target image (b) matching images using S_6 and Gödel implication (c) matching images using S_6 and Willmott implication

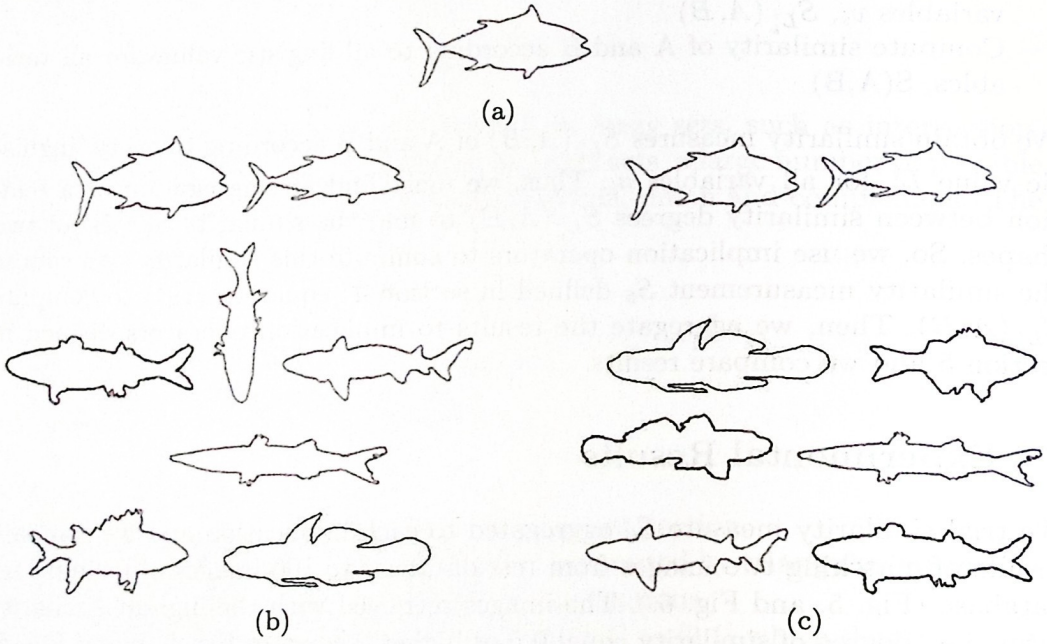


Fig. 6. (a) Target image (b) matching images using S_6 and Gödel implication (c) first's matching images using S_6 and Willmott implication

If we examine Fig. 5. and Fig. 6., we remark that the results produced with S_6 and Gödel implication are different from the results produced with S_6 and Willmot implication. To find the implication aggregated with S_6 which produces best results, we compute recall rate and precision rate that we present in table 2.

Table 2. Results obtained for each implication aggregated with S_6

Implication	Shape	Recall rate (%)	Precision rate (%)
Lukasiewicz		60%	75%
		57.14%	72.72%
Reichenbach		57,14%	100%
		61,53%	72.72%
Goguen		80%	100%
		61,53%	72.72%
Kleene-Dienes		57.14%	100%
		61,53%	72.72%
Gödel		80%	100%
		87,5%	63.63%
Rescher-Gaines		80%	100%
		87,5%	63,63%
Willmott		66,66%	100%
		75%	54,54%

We can observe from the table 2 that the best results are produced by S_6 aggregated with Goguen, Gödel or Rescher-Gaines implications.

8 Conclusion

Eventually, we presented a comparative study of aggregation implications to a fuzzy similarity applied to shape recognition. The experimental results show the difference between the implications aggregation. So, for the same shape sample, we found for the aggregation of Gödel implication to a fuzzy similarity a recall rate equal to 80% and a precision rate equal to 100% and for the aggregation of the Willmott implication to the fuzzy similarity a recall rate equal to 66,66% and a precision rate equal to 100%. Thus, we show with this comparison the

importance of the choice of the aggregation operator in order to compute the fuzzy similarity between many linguistic variables.

References

1. Wang, X., Bates, B., Kerre, E.: A comparative study of similarity measures. Fuzzy sets and systems **73** (1995) 259–268
2. Pappis, C., Karacapilidis, I.: A comparative assessment of measures of similarity of fuzzy values. Fuzzy sets and systems **56** (1993) 171–174
3. Mokhtarian, F., Abbasi, S., Kittler, J.: Efficient and robust retrieval by shape through curvature scale space. Proceedings of the first International Workshop on Image Databases and Multimedia Search, August (1996) 35–42
4. Mokhtarian, F., Abbasi, S., Kittler, J.: Robust and efficient shape indexing through curvature scale space. British Machine Vision Conference, September (1996) 53–62
5. Kopf, S., Haenselmann, T., Effelsberg, W.: Shape-based posture recognition in videos. Proceedings of Electronic Image, January **5682** (2005) 114–124
6. Zadeh, L.A.: Fuzzy sets. Inform and control **8** (1965) 338–356
7. Chen, S.M., Yeb., M.S., Hsiao, P.Y.: A comparison of similarity measures of fuzzy values. Fuzzy sets and systems **72** (1995) 79–89
8. Wang, W.: New similarity measures on fuzzy sets and on elements. Fuzzy sets and systems **85** (1997) 305–309
9. Ben Ayed, D., Ellouze, N.: Classification phonétique basée sur les mesures de similarité floues : Application aux signaux de parole de type voyelle. In: JTEA Conference, Sousse Nord Tunisie, Mars. (2002)
10. Bouchon-Meunier, B.: La logique floue et ses applications. SA (1995)
11. Dubois, D., prade, H.: Fuzzy criteria and fuzzy rules in subjective evaluation. In: A General Discussion. (1997)

Uncertainty Management in Context-Aware Applications

Pedro Damián-Reyes[1], Jesús Favela[1], Juan Contreras-Castillo[2], JRG Pulido[2]

[1] Centro de Investigación Científica y de Educación Superior de Ensenada, Km. 107 Carretera Tijuana-Ensenada, CP 22860, Ensenada, B. C., México
{preyes, favela}@cicese.mx
[2] Universidad de Colima, Facultad de Telemática, Av. Universidad #333, CP 28040, Colima, Col. México
{juancont, jrgp}@ucol.mx

Abstract. The objective of context-aware computing is to develop applications which perceive and analyze the user's activities and behavior to provide him with relevant information and services. However, the mechanisms used for perception or context analysis could provide uncertain information, generated by physical or logical factors. If this uncertainty is not properly managed, a context-aware application might become inadequate to the user. In this paper we propose a three-phase strategy to manage the uncertainty generated by contextual information within a context-aware application. Its objective is to implement actions to notify the user about the degree of uncertainty estimated and reduce it automatically through mechanisms such as re-estimation, recurrence, the use of additional contextual elements, or assisted via user feedback. We developed an activity-aware application aimed at a medical scenario to evaluate the effectiveness of the proposed strategy. The application simulates 7 hours of a nurse's work-day in a hospital. As a result we obtained that 61% of the elements marked as uncertain were due to an erroneous estimation of the algorithm. When we included automatic actions to reduce uncertainty this percentage grew to 76%. At the same time, when we included assisted actions the system requested user intervention on 13 times of all the entries in the registry.

1 Introduction

The rapid evolution of ubiquitous computing opens new possibilities to assist many activities. The health sector is a particularly appropriate area for the deployment of *ubicomp* applications, due to the fact that it is a highly dynamic environment with considerable worker mobility. Hospital workers carry out different activities at the same time, collaborate with other colleagues to solve problems, and use a wide variety of instruments and devices [1-3]. Recently, some hospitals have introduced wireless networks, PDA's, laptops, RFID labels, communication systems and some sensors for patient monitoring [4-7].

In a hospital environment workers need information highly dependent on contextual information such as location, role, time of day and activity [8]. This has motivated the development of context-aware applications to better assist hospital workers

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 77–86

Received 17/06/07
Accepted 31/08/07
Final version 22/10/07

adapting to the continuously changing environment [1, 8]. Some contextual information such as role and time of day can be easily determined, but other like location [9-13] or activity [14-16], require complex estimation algorithms and in some cases they use different technology and devices; which could generate uncertainty within an application, because the contextual information may be incomplete, erroneous or ambiguous. Uncertainty is an unavoidable factor in some context-aware applications. This generates one of the most important technical challenges in the development of context-aware computing: create mechanisms that provide applications the capacity to manipulate imperfect, uncertain or ambiguous contextual information [17, 18].

In this paper we propose a three-phase strategy to manage the uncertainty generated by contextual information within a context-aware application. Its objective is to implement actions to notify the user of uncertainty and reduce it: automatically through re-estimation, recurrence and combination with other contextual elements or additional information and assisted using user feedback. At the same time we applied this strategy to implement a mechanism for uncertainty management in an algorithm of activity estimation. This mechanism allows the classification of the estimates as certain or uncertain.

Finally, we tested and evaluated the uncertainty management mechanism in a medical scenario, in which we introduced an activity-aware application that simulates a 7 hours nurse's work-day in a hospital.

The rest of the paper is organized as follows: Section 2 presents previous related works. In section 3, a description of the suggested strategy for uncertainty management is presented. The algorithm used to estimate the activity is described in section 4, and the implementation of the strategy is presented in section 5. The outcomes of the application are presented in section 6. Conclusions are described in section 7.

2 Related Work

Bayesian networks are a powerful tool to deal with uncertain information. Tau Gu et al. [19] used Bayesian networks to estimate the activity of an user. They use RDF ontologies to define context, they later transform the ontology into a Bayesian network. Ranganathan et al. [20] implemented a reasoning engine where they combined rules and information inference to influence decisions using Bayesian networks. Truong et al. [21] also use Bayesian networks and ontologies, but their focus is on reusing context ontology definitions. They outline an ontology structure that may be used in different scenarios, helping decrease the time and effort required for the definition of context. The proposals above address the problem of dealing with uncertain information using probabilistic reasoning. However, a great amount of work by application developers is required and the aid of experts to create the context ontology. Also, the transformation of the ontology to a Bayesian network is not automatic and it is a very demanding task. Our proposal is based on an automated low-cost strategy based on a three-phase structure consisting of a context-aware mechanism to identify the presence of uncertainty and to define actions to carry out in the presence of such uncertainty based on specific rules. The strategy can be used in the design of a new

context-aware application or to implement additional mechanism that enhances an existing application.

3 Strategy for Uncertainty Management

Figure 1 depicts our uncertainty management strategy for context-aware applications. The first step consists on the identification of uncertainty, the second step consists on measuring the degree of uncertainty, and the third step consists on adopting actions to attenuate the uncertainty.

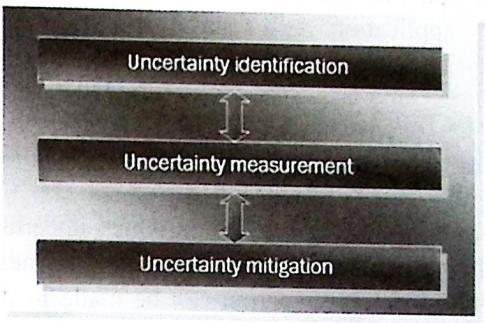


Figure 1. Uncertainty management strategy

3.1 Identification of Uncertainty

A detailed analysis of the application and its relationship with the contextual variables is conducted at this stage, with the aim of identifying conditions in which the uncertainty appears.

To identify uncertainty, it is convenient to define a general model of the mechanism of context-awareness (Fig.2). This structure consists of two main elements: inputs and processes. The former are data and contextual elements are considered inputs. The latter are algorithms, rules, reasoning methods, procedures and actions that provide awareness to the application.

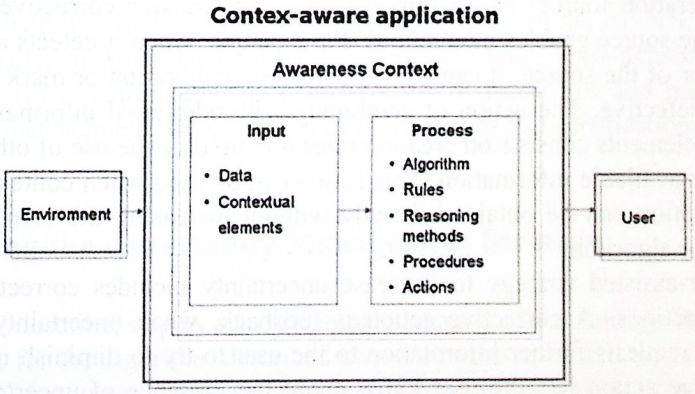


Figure 2. General model of context aware mechanisms

It is possible to analyze in an independent way, and determine where the uncertainty may appear (Fig.2). In some situations the uncertainty may appear in the inputs, while in others it might appear in the processes, or in both.

Inputs with uncertainty are associated to contextual elements. When the source of generation of contextual element is known, it is possible to better analyze its operation, in search of behavioral patterns that later on can be used to dimension the level of uncertainty.

Processes with uncertainty are associated to the way in which the application becomes aware of the context. This is how inputs are used to provide awareness to the application. We must analyze how contextual information is used and what actions are carried out with that information. This includes rules, reasoning methods and processes used by the application. As a result of this stage the designer identifies the sources of uncertainty in the mechanism of context awareness.

3.2 Measurement of Uncertainty

This phase consists on establishing the way in which uncertainty is represented and quantified: finding the way of making samples tangible or measurable. Depending on the kind of uncertainty, the representation can be made in different ways, in some situations specific values are used, such as true/false, percentages, probabilities, averages, distances, among other, and in other cases, the representations can be better defined with words, for these cases it is possible to use fuzzy logic. In this stage it is expected as a result, a series of rules or diagnoses tests to identify the presence or absence of uncertainty.

3.3 Actions for Dealing with Uncertainty

It consists on establishing actions the application must carry out in the presence of uncertainty to mitigate or control its effects.

To decreased uncertainty, corrective actions can be used by the application. These actions include: re-estimation. This consists on making a new estimation of the contextual element. It is executed when there is a doubt about the information provided by the generation source. This action can be used to make a corrective action to the source if the source enables interactions. For example, when it detects an inappropriate behavior of the source, it can try to solve it by calibration or mark the device or sensor as defective. The action of combining with additional information or others contextual elements consists on creating rules that involve the use of other contextual elements or available information. This action can be used when contextual elements and information can be obtained directly without increasing the complexity of the procedure or algorithm.

The user-assisted strategy to decrease uncertainty includes corrective and non-corrective actions. A corrective action is feedback, when uncertainty appears the application requests further information to the user to try to diminish uncertainty; A noncorrective action lets the user know about the presence of uncertainty, and the user must decide on the appropriate course of action. These actions can be conducted

through adaptation of the user interface, with use of text, colors, icons, sounds and combinations of them.

4 The Estimation of a User's Activity

A procedure to estimate user activity for context-aware applications is proposed in [22]. In this approach, a back propagation neural network (BP-NN) is used to estimate the user activity by mapping from contextual variables to activities. The BP-NN receives as input user's contextual variables, such as its current location, the time of day, artifacts being used and the presence of colleagues. The neural net maps this inputs to output values corresponding to every possible activity performed by the worker. The output with highest activation value indicates the activity that the worker is carrying out at that moment. The structure of the BP-NN used is illustrated in Figure 3. This network uses, as inputs, 1 neuron for time of the day, 1 neuron for user location, 1 neuron for every significant artifact the worker may use and, 1 neuron for the role of each person with whom they may collaborate; and, as outputs, 1 neuron for every possible activity performed by the worker.

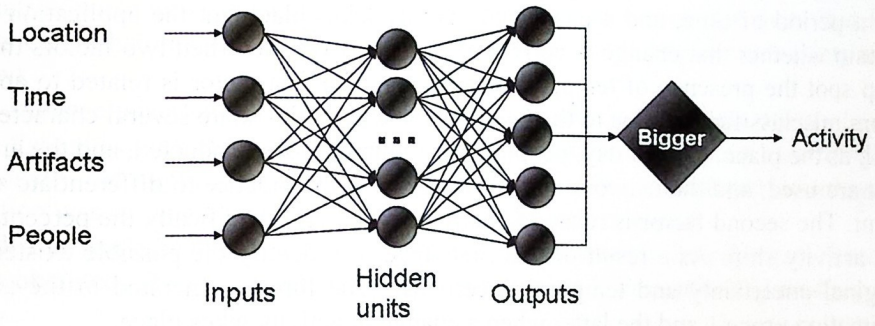


Figure 3. Structure of BP-NN used to estimate user's activity

The authors based their approach on information gathered from a workplace study conducted in a hospital, in which 196 hours of detailed observation of hospital workers were recorded. The record structure includes location, time of day, artifacts, people and real activity. These records are used to train and test the BP-NN. They define 7 main activities in which hospital workers engage: clinical case assessment (CCA), patient care (PC), coordination (C), preparation of medicines and material (PM), information management (IM), tracking (T) and, classes and certifications (CC).

5 Applying the Uncertainty Management Strategy

Now we will describe, step by step, the implementation of our strategy for uncertainty management for the algorithm described above. For this, we used the same data collected in [29] for nurses. These records were created as result of direct observation at the hospital.

5.1 Identification of Uncertainty

In this first step, we identify the existence of uncertainty. We define uncertainty in a descriptive way. As mentioned before, the procedure is based on the use of a neural network to estimate the user activity by mapping from contextual variables to activities. The estimate activity corresponds to the output neuron with highest activation value. Secondly, we define *original uncertainty*. This uncertainty appears for a single moment and a single estimate. Then, when we estimate an activity from the algorithm the original uncertainty appears, since it is likely that the estimate made is incorrect.

We identified three factors that can help spot the presence of uncertainty. The first factor is related to the nature of the neural network, this is when the outputs values of each neuron are very similar among them. The next two factors are related to the contextual information used, that is, the inputs to the neural network. We observed that each activity performed by a hospital worker depends on contextual variables such as user location, the people with whom they collaborate and the artifacts used during the execution of the activity. For example, classes and certification is an activity executed in base locations, such as meeting rooms or offices, and when an instructor is present.

Let us now define *temporal uncertainty*. This appears when the application is used for a period of time, and a change of activity takes place but the application is not certain whether that change is correct or incorrect. We identified two factors that can help spot the presence of temporal uncertainty. The first factor is related to an algorithm misclassification due to the fact that some activities share several characteristics such as the place, time of day, people with whom they are conducted, and the artifacts that are used, and the network does not have enough evidence to differentiate among them. The second factor is related to activity switching, specifically the percentage of the activity shift. As a result of this first stage, we identify the possible existence of original uncertainty and temporal uncertainty. The former generated in the activity estimation process and the latter when a change of activity takes place.

5.2 The Measurement of Uncertainty

The second step consists on measuring the degree of uncertainty. The uncertainty must be represented to create diagnostic tests to determine whether the uncertainty is significant to act on it.

In medicine a probabilistic process of validation to evaluate the efficient of a diagnostic test is commonly used [23, 24]. The process includes sensibility, specificity, positive and negative predictive values. We used that process to avoid the creating arbitrary rules. We apply the validation process using different values to obtain the correct value that was used in the rule.

To indicate the presence of uncertainty one rule was established: this rule indicates the presence of uncertainty when the difference between the biggest output value and the previous one are smaller or equal to 0.1 (Fig.4).

5.3 Actions for Dealing with Uncertainty

At this stage, the actions that will be carried out in the presence of uncertainty are established. To automatically decrease uncertainty (Fig.5), re-estimation, does not apply in our case, because the inputs, as mentioned above, are obtained from a record generated by direct observation, so we can assume that these values are free from error. In the combination of additional information or others contextual elements, a first action may be a procedure to look for the nearest nurse from the current location. It uses the same activity for this nurse, in a real implementation it is possible to use this process because oftentimes there are several nurses working at the same place, at the same time. For our simulation this action was not applied because the data used corresponds to a single nurse. The second action is recurrence which is based on the principle that a nurse starts an activity and does not change to another one until the first one has been concluded. In this action it is assumed that the correct activity is the one done prior to he detection of uncertainty. The third action uses nurse location and estimates whether it is uncertain. If so, it verifies the location and if the nurse is in a place where it is feasible to undertake such activity, the uncertain status disappears and assumes that it is a correct estimate, otherwise the estimate continues being uncertain and proceeds to inform the user. The fourth action involves the activity transition matrix, when an estimate is uncertain it is assumed as a correct activity that has the highest probability of change based on the activity made before uncertainty was detected.

To decrease uncertainty based on user-assisted feedback, when uncertainty is detected, the application explicitly inquires to the nurse "What activity are you doing?". Additionally the application can inform the user about the presence of uncertainty, this action was conducted with the use of color in the user interface.

With the implementation of these strategies an uncertainty management mechanism is obtained. The general structure of the mechanism is present in Figure 4.

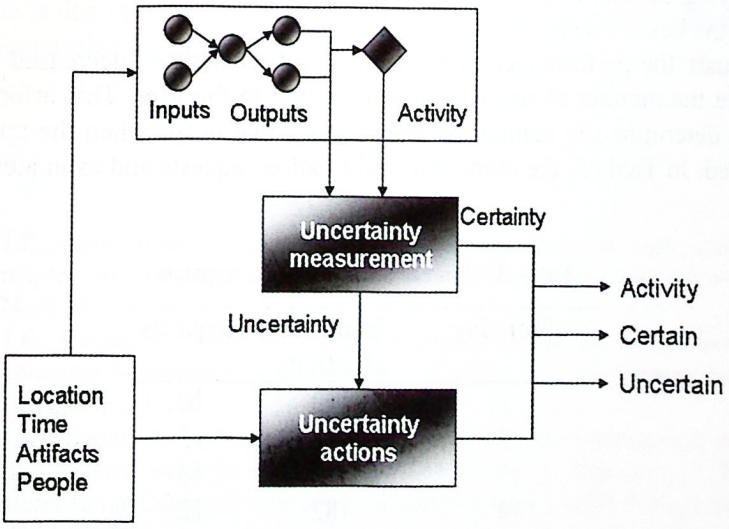


Figure 4. General structure of uncertainty management mechanism

6 Test and Results

In order to evaluate our uncertainty management approach we conducted a five day simulation of a nurse workday, form 7:00 AM to 2:30 PM. Data taken from the observation logs were used for this purpose. The simulation consists of the following steps: the BP-NN is fed with data from the logs and an output is obtained. This output is an estimate we use to compare to the real activity, allowing us to know whether or not the estimate is right.

The metric established to evaluate the performance of the actions to automatically decrease uncertainty consists on measuring the number of uncertainty estimates that correspond to an estimation error. Table 2 shows the percentages of uncertain estimates that correspond to errors by the algorithm, the first column corresponds to the classifier, the second to the implementation of recurrence action, the third to the use of location, and fourth to the use of a transition matrix.

Table 2. Percentage of uncertain estimates that corresponds to estimation errors

Workday	Classifier	Recurrence	Location	Transition matrix
1	60%	86%	100%	60%
2	60%	60%	50%	75%
3	56%	69%	67%	60%
4	75%	75%	100%	75%
5	54%	88%	50%	54%
Average	61%	76%	73%	65%

As can be seen from Table 2, 61% of the estimates marked as uncertain correspond to incorrect estimations made by the algorithm. At the same time we can see that when applying actions to decreased uncertainty the percentage increases to 76% on average in the best of cases.

To evaluate the performance of the actions to user-assisted decreased uncertainty we measure the number of requests of information to the user. This information can be used to determine the number of interruptions that occur when the application is implemented. In Table 3, the number of information requests and estimates in the day is shown.

Table 3. Number of information requests

Workday	Estimates of activity	Requests
1	170	10
2	175	15
3	259	16
4	182	12
5	192	13
Average	196	13

As can be seen from Table 3, when applying the actions of the user-assisted decreased uncertainty the application informs the user or requested feedback on 13 times of 196 estimates of activity on average. It is important to consider this value due to the fact that the request may become an unwanted interruption and jeopardize the usefulness of the application.

7 Conclusions

This paper presents a strategy for uncertainty management based on a three-phase structure: identification, measurement and dealing with uncertainty. This strategy proposes a series of actions that must be considered when designing context-aware applications that need to deal with uncertain information. These actions are: the automatic decrease of uncertainty which includes actions such as re-estimation, recurrence and combination with additional information or others contextual elements, and the user-assisted decreased uncertainty with actions such as obtaining user feedback and keeping users informed about the presence of uncertainty. The strategy can be applied not only to new applications but also to enhance existing ones.

As a result of evaluation of the proposed strategy we observed that uncertainty management can be useful to design context-aware applications. We established a metric to evaluate the utility of decrease uncertainty actions based on an algorithm to minimize error, even when error is not a synonymous of uncertainty. With this metric, it was observed that 61% of the estimates marked as uncertain correspond to erroneous estimates. After including auto-decreased uncertainty actions the utility was increased, because the percentage of the estimates marked as uncertain correspond to erroneous estimates increased by 76% on average. On the other hand, when actions to decreased user-assisted uncertainty are applied, the application informs the user or requests feedback on 13 times of 196 estimates of activity. It is important to consider that this value is due to the fact that the request may itself become an unwanted interruption and jeopardize the application usefulness.

References

1. Bardram, J.E.: Applications of context-aware computing in hospital work: examples and design principles. ACM symposium on Applied computing. ACM Press, Nicosia, Cyprus (2004) 1574-1579
2. Bardram, J.E., Bossen, C.: Mobility Work: The Spatial Dimension of Collaboration at a Hospital. Computer Supported Cooperative Work (CSCW), Vol. 14. Springer Netherlands, Paris, France (2005) 131-160
3. Bossen, C.: The parameters of common information spaces: the heterogeneity of cooperative work at a hospital ward. In: Churchill, E.F., McCarthy, J., Neuwirth, C., Rodden, T. (eds.): Conference on Computer supported cooperative work. ACM Press, New Orleans, Louisiana, USA (2002)
4. Chin, T.: Untapped power: A physician's handheld. American Medical News (2005)
5. O'Connor, M.C.: Testing Ultrasound to Track, Monitor Patients. RFID 1 (2006) 2

6. Pentland, A.: Healthwear: Medical Technology Becomes Wearable. IEEE Computer, Vol. 37 (2004) 34-41
7. Stanford, V.: Beam Me Up, Doctor McCoy. Pervasive Computing, Vol. 2 (2003) 13-18
8. Muñoz, M.A., Rodríguez, M., Favela, J., García, A.I.M.: Context-aware mobile communication in hospital. IEEE Computer, Vol. 36 (2003) 38-46
9. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. 6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM). ACM Press, Boston, MA (2000) 32-43
10. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems, Vol. 10 (1992) 91-102
11. Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based location and tracking system. IEEE INFOCOM 2000, Vol. 2, Tel-Aviv, Israel (2000) 775-784
12. Rodríguez, M.D., Favela, J., Martínez, E.E., Muñoz, M.A.: Location-Aware Access to Hospital Information and Services. IEEE Transaction on Information Technology in Biomedicine, Vol. 8 (2004) 448-455
13. Castro, L.A., Favela, J.: Continuous Tracking of User Location in WLANs Using Recurrent Neural Networks. Sexto Encuentro Internacional de Computación (ENC 2005). IEEE Press, Puebla, Mexico (2005) 174-181
14. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Fox, D., Kautz, H., Hähnel, D.: Inferring activities from interactions with objects. IEEE Pervasive, Vol. 3 (2004) 10-17
15. Bian, X., Abowd, G.D., Rehg, J.M.: Using Sound Source Localization to Monitor and Infer Activities in the Home. GVU Technical Report; GIT-GVU-04-20. Georgia Institute of Technology (2004)
16. Christensen, H.B., Bardram, J.: Supporting Human Activities - Exploring Activity-Centered Computing. 4th international conference on Ubiquitous Computing, Vol. 2498. Springer-Verlag, Göteborg, Sweden (2002) 107-116
17. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications **8** (2001) 10-17
18. Satyanarayanan, M.: Coping with Uncertainty. IEEE Pervasive Computing **2** (2003) 2
19. Gu, T., Pung, H.K., Zhang, D.Q.: A Bayesian Approach for Dealing with Uncertain Contexts. 2nd Intl Conf on Pervasive Computing (Pervasive 2004), Vol. 176, Austria (2004)
20. Ranganathan, A., Al-Muhtadi, J., Campbell, R.H.: Reasoning about Uncertain Contexts in Pervasive Computing Environments. IEEE Pervasive Computing Journal **3** (2004) 62-70
21. Truong, B.A., Lee, Y.-K., Lee, S.-Y.: Modeling and Reasoning about Uncertainty in Context-Aware Systems. IEEE International Conference on e-Business Engineering (ICEBE'05). IEEE Computer Society, Beijin, China (2005) 102-109
22. Favela, J., Tentori, M., Castro, L.A., Gonzalez, V.M., Moran, E.B., Martínez-García, A.I.: Activity Recognition for Context-Aware Hospital Applications: Issues and Opportunities for the Deployment of Pervasive Networks. MONET (2007)
23. Altman, D., Bland, J.: Statistics Notes: Diagnostic tests 1: sensitivity and specificity. BMJ 6943 (1994) 1552
24. Altman, D., Bland, J.: Statistics Notes: Diagnostic tests 2: predictive values. BMJ 6947 (1994) 102

Knowledge Representation and Reasoning

Basic Transformations that Preserve G'3-stable Equivalente

José Luis Carballido[1], Mauricio Osorio[2], and José Arrazola[1]

[1] Benemérita Universidad Autóma de Puebla,
Mathematics Department, Puebla, México
carballido, arrazola@fcfm.uap.buap.mx
[2] Universidad de las Américas,
Sta. Catarina Mártir, Cholula, Puebla, México
osoriomauri@gmail.com

Abstract We study the notion of strong equivalence between two disjunctive logic programs under the G'_3-stable model semantics, also called the P-stable semantics, and we show how some particular cases of testing strong equivalence between programs can be reduced to verify if a formula is a theorem in some paraconsistent logics or in some cases in classical logic. We also present some program transformations for disjunctive programs which can be used to reduce the size of a program.

1 Introduction

In [11,12], a new semantics for non-monotonic reasoning was introduced in terms of weak completions of the three-valued G'_3-logic. It is known that this semantics, as well as semantics defined with the same construct by other paraconsistent logics, is equivalent to the P-stable model semantics for normal programs [12]. In [8] the result is extended to the class of disjunctive programs. The P-stable semantics of a normal or disjunctive program P is defined by means of a fixed point operator in terms of classical logic, after applying to P a transformation similar to the reduction used to define the stable semantics [4]. Similar results for disjunctive programs are presented in [13] for the more common stable semantics. The stable semantics turns out to be equivalent, for disjunctive programs, to semantics defined in terms of weak completions of any intermediate logic. In [8] the authors present a simple translation of a disjunctive program D into a normal program N, such that the P-stable semantics of N corresponds to the stable semantics of D over the common language.
The logic G'_3 was introduced in [1] only to prove that the formula $A \vee (A \rightarrow B)$ is not a theorem of the logic C_w. In fact, the set of theorems of G'_3 is a superset of the set of theorems of C_w. One interesting feature of the G'_3-logic is that it can be expressed in terms of the Lukaciewicz L_3-logic, and vice-versa, the Lukaciewicz L_3-logic can be expressed in terms of the G'_3-logic [9]. In the same survey it is proven that the logic G'_3 admits a finite axiomatization, in fact the axiomatization presented there, consists of the axioms for Cw plus four new

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 89–98

Received 17/06/07
Accepted 31/08/07
Final version 21/09/07

axioms.
Two programs P_1 and P_2 are said to be strongly G'_3-stable equivalent, if for every program P, the programs $P_1 \cup P$ and $P_2 \cup P$ are G'_3-stable equivalent, i.e. they have the same G'_3-stable models. The notion of strongly equivalent logic programs is interesting since, given two sets of rules that are strongly equivalent, one of them can be replaced by the other one in any logic program without changing the declarative semantics of the program. This replacement can be a step toward program simplification. The notion of equivalence between logic programs has been studied by several authors in the context of the stable semantics, for example [3,6,10]. In the present work a different kind of equivalence is considered along with strong G'_3-stable equivalence for disjunctive programs, it is called uniform G'_3-stable equivalence and is weaker than strong G'_3-stable equivalence. G'_3 is one of several paraconsistent logics that can express the p-stable semantics for disjunctive programs, as defined in [8]. Therefore conditions under which two programs are strongly G'_3-equivalent also guarantee that two disjunctive programs are strongly equivalent in the p-stable semantics. We present two main results that guarantee G'_3 strong equivalence, one for two arbitrary programs and another one for a couple of programs of the form P , $P \cup \{a\}$, where P is a disjunctive program and a is an atom. A similar result for the stable semantics is presented in lemma 4.2 of [10] for disjunctive programs, but in this case the necessary condition for the strong equivalence of P and $P \cup \{a\}$ is more stringent. Both of our results depend on verifying that certain formula is a theorem in some particular logic. Finally, we present some program transformations that help reduce the size of a program. In most cases the transformed program is strongly G'_3-equivalent to the original program.
The structure of the paper is as follows: we start with basic background and definitions of the G'_3-logic, the P-stable semantics and the X-stable semantics for any logic X. Section 3 presents first the main results relative to strong equivalence of two disjunctive programs, and then it deals with some basic transformations of programs. Then we present our conclusions and ideas for future work.

2 Background

A *signature* $\mathcal{L}$ is a finite set of elements that we call *atoms*, or *propositional symbols*. The language of a propositional logic has an alphabet consisting of

proposition symbols: $p_0, p_1, \ldots$
connectives: $\wedge$, $\vee$, $\leftarrow$, $\neg$ auxiliary symbols: (,).

Where $\wedge$, $\vee$, $\leftarrow$ are 2-place connectives and $\neg$ is a 1-place connective. Formulas are built up as usual in logic. If F is a formula we will refer to its signature $\mathcal{L}_F$ as the set of atoms that occur in F. The formula $F \equiv G$ is an abbreviation for $(F \leftarrow G) \wedge (G \leftarrow F)$. A *literal* is either an atom a, or the negation of an atom $\neg a$.

When a formula is constructed as a conjunction (or disjunction) of literals. $F = \bigwedge \ell$ (or $F = \bigvee \ell$) with ℓ a set of literals, we denote by $Lit(F)$ such set of

literals ℓ. A *clause* is a formula of the form $H \leftarrow B$ where H and B, arbitrary formulas in principle, are known as the *head* and *body* of the clause respectively. The body of a clause could be empty in which case the clause is known as a *fact* and can be noted just by H.

An *augmented* clause is a clause where H and B are any formulas constructed by using the $\vee, \wedge, \neg$ connectives. A *free* clause is a clause of the form $\bigvee(\mathcal{H}^+ \cup \neg\mathcal{H}^-) \leftarrow \bigwedge(\mathcal{B}^+ \cup \neg\mathcal{B}^-)$ where $\mathcal{H}^+$, $\mathcal{H}^-$, $\mathcal{B}^+$, $\mathcal{B}^-$ are, possibly empty, sets of atoms. Sometimes such a clause may be written as $\mathcal{H}^+ \vee \neg\mathcal{H}^- \leftarrow \mathcal{B}^+, \neg\mathcal{B}^-$ following typical conventions for logic programs. When $\mathcal{H}^- = \emptyset$, there is no negation in the head, the clause is called a *general* clause. If, moreover, $\mathcal{H}^+ \neq \emptyset$ (i.e. it not a constraint) the clause is called a *disjunctive* clause. When the set $\mathcal{H}^+$ contains exactly one element the clause is called *normal*.

Finally, a *program* is a finite set of clauses. If all the clauses in a program are of a certain type we say the program is also of this type. For instance a set of augmented clauses is an *augmented program*, a set of free clauses is a *free program* and so on.

For general programs, and proper subclasses, we will use $HEAD(P)$ to denote the set of all atoms occurring in the head of clauses in P.

Next, we proceed to give definitions of the relevant semantics and the three-valued logic G'_3

2.1 P-stable Semantics

We will use the following notation: $\models$ denotes the consequence relation in classical logic. We denote by $\vdash_{C_w}$ the inference relation in logic C_w which is the minimal paraconsistent logic defined by DaCosta [2]. We also assume that the reader is familiar with the notion of classical minimal model [7].

Here we define the P-stable semantics for disjunctive programs.

Definition 1. *[8] Let P be a disjunctive program and M be a set of atoms. We define:*

$$RED(P, M) = \{H \leftarrow B^+, \neg(B^- \cap M) \mid H \leftarrow B^+, \neg B^- \in P\}$$

Definition 2. *[8] Let P be a disjunctive program and M be a set of atoms. We say that M is a p-stable model of P if the conjunction of the atoms in M is a logical consequence in classical logic of $RED(P, M)$ (denoted as $RED(P, M) \models M$) and M is a classical model of P (i.e. a model in classical logic).*

2.2 G'_3 Logic

G'_3 logic is defined through a 3-valued logic with truth values in the domain $D = \{0, 1, 2\}$ where 2 is the designated value. The evaluation function of the logic connectives is then defined as follows: $x \wedge y = \min(x, y)$; $x \vee y = \max(x, y)$; and the $\neg$ and $\rightarrow$ connectives are defined according to the truth tables given in Table 1.

x	$\neg x$
0	2
1	2
2	0

$\rightarrow$	0	1	2
0	2	2	2
1	0	2	2
2	0	1	2

Table 1. Truth tables of connectives in G'_3.

2.3 The $\dot{X}$-stable Semantics

Given an arbitrary program P and a set of atoms $M \subset L_P$, we call the construct $P \cup \neg M^c$ a *weak completion* of the program P (with respect to the set of atoms M), where the superscript c, denotes set theoretical complement operator with respect to L_P.

Definition 3. *Let P be any theory and X be any logic. Also let M be a set of atoms. M is a X-stable model of P if $P \cup \neg M^c \vdash_X M$ and M is a classical model of P.*

The G'_3-stable semantics Of particular interest to us is the G'_3-stable semantics, which is the result of using the logic G'_3 in the previous definition.

Example 1. Consider the following logic program:

$$P = \{b \leftarrow \neg a, a \leftarrow \neg b, p \leftarrow \neg a, p \leftarrow \neg p\}$$

It is easy to verify that this program has two G'_3-stable models, which are $\{a, p\}$ and $\{b, p\}$.

The next result was first proven for normal programs in [12]; more recently it has been extended to disjunctive programs in [8].

Theorem 1. *[8] Let P be a disjunctive program and M be a set of atoms. M is a p-stable model of P iff M is a G'_3-stable model of P.*

As can be seen, G'_3-stable models are defined for propositional logic programs only. However this definition can be extended to *predicate programs*, which allow the use of predicate symbols in the language, but without function symbols to ensure the ground instance of the program to be finite. So a *term* can only be either a variable or a constant symbol. The ground instance of a predicate program, $Ground(P)$, is defined in [5] as the program containing all ground instances of clauses in P. Then M is defined as a G'_3-stable model of a predicate program P if it is a G'_3-stable model for $Ground(P)$.

We want to stress the fact that the general approach for calculating $G3'$-stable models of logical programs is to work with their ground instances.

3 Main Results

In this section we discuss the main contributions of the paper, one of them provides sufficient conditions for the strong G'_3-stable equivalence of two arbitrary programs, the other one deals more specifically with a couple of programs of the form P and $P \cup \{a\}$.

Definition 4. *We denote by C the class of certain programs i.e. normal, disjunctive, positive, augmented, etc. We say that two programs P_1 and P_2 in C are G'_3-stable equivalent if they have the same G'_3-stable models. We say that the programs P_1 and P_2 in C are strongly G'_3-stable equivalent with respect to the class C, if for any other program $P \in C$, $P \cup P_1$ and $P \cup P_2$ are G'_3-stable equivalent. We say that the programs P_1 and P_2 are uniformly G'_3-stable equivalent if for any set of atoms M, $P_1 \cup M$ and $P_2 \cup M$ are G'_3-stable equivalent.*

Proposition 1. *Let P_1 and P_2 be programs in the same class C. If P_1 and P_2 are strongly G'_3-stable equivalent with respect to the class C then they are uniformly G'_3-stable equivalent. If P_1 and P_2 are uniformly G'_3-stable equivalent then they are G'_3-stable equivalent.*

Proof. The proof of this statement is a direct consequence of the definitions.

The next two examples show that two programs that are G'_3-stable equivalent are not necessarily uniformly G'_3-stable equivalent. Our first counterexample is with normal programs:

Example 2.

$$P_1 = \{a \leftarrow \neg b, \quad b \leftarrow b\}$$

$$P_2 = \{a, \ b \leftarrow b\}$$

Both programs have as unique G'_3-stable model the set: $\{a\}$.
If we take

$$P = \{b\},$$

then we see that $P \cup P_1$ has only one G'_3-stable model, namely: $\{b\}$, whereas $P \cup P_2$ has as its unique G'_3-stable model the set $\{a, b\}$.
As a second counterexample we take two programs in the class of disjunctive programs:

$$P_1 = \{a \vee c \leftarrow \neg b\}, P_2 = \{a \vee c\}.$$

These programs do not have G'_3-stable models, but if we add the fact b, to both of them, we get a program for which the set $\{b\}$ is a G'_3-stable model and a program that does not have G'_3-stable models.

Proposition 2. *Let P_1 and P_2 be arbitrary programs, if $P_1 \equiv_{G'_3} P_2$ (i.e. they are equivalent in the G'_3 logic), then they are strongly G'_3-stable equivalent with respect to the class of arbitrary programs.*

Proof. Let P be another arbitrary program, let M be a G'_3-stable model of $P_1 \cup P$, this means that M is a classical model of $P_1 \cup P$ and also that $P_1 \cup P \cup \neg M^c \vdash_{G'_3} M$. Since $P_1 \equiv_{G'_3} P_2$, it follows that $P_1 \cup P \equiv_{G'_3} P_2 \cup P$ and then we conclude that M is a classical model of $P_2 \cup P$ and $P_2 \cup P \cup \neg M^c \vdash_{G'_3} M$ as desired.

A symmetric argument shows that a G'_3-stable model of $P_2 \cup P$ is also a G'_3-stable model of $P_1 \cup P$.

The converse of the proposition in general is not true. We consider a counterexample in the class of disjunctive programs.

Example 3. Let P_1 and P_2 be the following disjunctive programs:

$$P_1 = \{\vee c_i \leftarrow \wedge a_j \wedge \neg b, \quad b \leftarrow\}$$

$$P_2 = \{\vee c_i \leftarrow c_1, \quad b \leftarrow\}$$

Let us see first that P_1 and P_2 are not equivalent in the G'_3-logic: in order to do this, it is enough to show that the two rules in P_2 are not strong enough in the G'_3-logic to imply the first rule in P_1. In other words, $[(c_1 \rightarrow \vee c_i) \wedge b] \rightarrow [(\wedge a_j \wedge \neg b) \rightarrow \vee c_i]$ is not a G'_3 tautology. This can be seen by working out a truth table.

Now we proceed to check that the two programs are strongly G'_3-stable equivalent. Let P be a disjunctive program, we want to see that $P_1 \cup P$ and $P_2 \cup P$ have the same p-stable models (by Theorem 1).

Let M be a p-stable model of $P_1 \cup P$, in particular it must be the case that $b \in M$. By definition, M is a classical model of $P_1 \cup P$ and also $RED(P_1 \cup P, M) \models M$, this last condition is equivalent to: $P_1 \cup RED(P, M) \models M$.

Now, from the fact that the expression: $b \rightarrow [(\neg b \wedge A) \rightarrow C]$ is a tautology in classical logic for any formulas A and C, it follows that $P_2 \models P_1$. It is also true that $P_1 \models P_2$ since the rule in P_2 that does not appear in P_1 is a tautology.

It follows that M is a classical model of $P_2 \cup P$ and that $RED(P_2 \cup P, M) \models M$ since $P_2 \cup RED(P, M) = RED(P_2 \cup P, M)$. The rest of the proof consist of a similar argument.

Observation 1: The same example works out as a counter-example for the class of normal programs if we remove the disjunctions in the heads of the disjunctive rules.

Observation 2: Notice that if we do not restrict our programs to the class of disjunctive programs, the counterexample does not work: consider the program $Q = \{d \leftarrow (\vee c_i \leftarrow \wedge a_j \wedge \neg b)\}$ consistent of a single clause. The two programs $P_1 \cup Q$ and $P_2 \cup Q$ do not have the same G'_3-stable models, so in this context P_1 and P_2 are not strongly G'_3-equivalent.

Lemma 1. *Let P be a disjunctive program and M be a classical model for P, then $P \cup M$ has G'_3-stable models.*

Proof. By hypothesis M is a classical model for P, then it is also a classical model for $P \cup M$. Since $RED(P \cup M, M) = RED(P, M) \cup M$, it is clear that $RED(P \cup M, M) \models M$. Hence M is a G'_3-stable model of $P \cup M$.

For the next result we need the following fact [8].

Lemma 2. *For a disjunctive program P, and an atom a we have that $P \models a$ if and only if $P \vdash_{C\omega} a$.*

Proposition 3. *Let P be a disjunctive program and a be an atom, then $P \models a$ if and only if P and $P \cup \{a\}$ are strongly G'_3-stable equivalent, if and only if P and $P \cup \{a\}$ are uniformly G'_3-stable equivalent.*

Proof. From $P \models a$. it follows that $P \vdash_{C\omega} a$ and from the fact that the G'_3-logic is stronger than the $C\omega$-logic, it follows that $P \vdash_{G'_3} a$. Hence we conclude that $P \equiv_{G'_3} P \cup \{a\}$, this in turn implies by proposition 2 that P and $P \cup \{a\}$ are strongly G'_3-stable equivalent and in particular, uniformly G'_3-stable equivalent. Conversely, let us assume now that the two programs are uniformly G'_3-stable equivalent. Assume for a moment that $P \not\models a$: then there exists a classical model of P, say M such that $M(a) = 0$. By Lemma 1, M is a G'_3-stable model of $P \cup M$, but M is not a G'_3-stable model of $P \cup \{a\} \cup M$. It means that P and $P \cup \{a\}$ are not uniformly G'_3-stable equivalent, but this is a contradiction and the proof is complete.

Definition 5. *Given a disjunctive program P, we define*

$$to - pos(P) = \{\bigvee_i a_i \vee \bigvee_j c_j \leftarrow \wedge_l b_l \mid \vee_i a_i \leftarrow \bigwedge_l b_l \wedge \bigwedge_j \neg c_j \in P\}.$$

The transformation just moves negative literals from the body of every clause to their heads by changing them to positive.

Corollary 1. *Let P be a disjunctive program and a be an atom. Then $to - pos(P) \vdash_{C\omega} a$ if and only if P and $P \cup \{a\}$ are strongly G'_3-stable equivalent, if and only if P and $P \cup \{a\}$ are uniformly G'_3-stable equivalent.*

Proof. Follows from the fact that $to - pos(P)$ and P are equivalent in classical logic, then by using Lemma 2 the result is simply a restatement of Proposition 3.

Definition 6. *Given a disjunctive program P. $Pos(P)$ is the program obtained from P after deleting all rules containing negative atoms.*

In [10] it is shown that for a disjunctive program P and an atom a, P is strongly equivalent to $P \cup \{a\}$ under the stable semantics if and only if $Pos(P) \vdash_I a$ (here the I denotes Intuitionism). In the case of G'_3-semantics the equivalence does not hold. From Proposition 3 it follows that if $Pos(P) \vdash_I a$ then P and $P \cup \{a\}$ are strongly G'_3-stable equivalent. The next examples show that the converse is not true.

Example 4. Let P_1 be: $\{a \leftarrow \neg a\}$ and let P_2 be: $P_1 \cup \{a\}$.
Since $(\neg a \rightarrow a) \rightarrow a$ is a G'_3-tautology, it follows that $P_1 \equiv_{G'_3} P_2$ which implies strong G'_3-stable equivalence, however $Pos(P_1) = \emptyset$ and from the empty set no atom can be derived.

Example 5. Let P_3 and P_4 the following disjunctive programs:

$$P_3 = \{a \leftarrow b, \quad a \leftarrow \neg b, \quad b \leftarrow a\}$$

$$P_4 = \{a, \quad a \leftarrow b, \quad a \leftarrow \neg b, \quad b \leftarrow a\}$$

The two programs are equivalent in the G'_3-logic, therefore they are strongly G'_3-stable equivalent, but the atom a is not implied by the rules of $Pos(P_3)$.

Finally we present some transformation defined for disjunctive programs that can help reduce the size of the program, most of them do not change the G'_3-stable semantics of the original program. Similar results in the context of stable semantics have been presented in [10].

Definition 7. *The transformation RED^+ replaces a rule $A \leftarrow B^+ \wedge \neg B^-$ by $A \leftarrow B^+ \wedge \neg(B^- \cap Head(P))$.*

Proposition 4. *For a disjunctive program P, P and $RED^+(P)$ are equivalent under the G'_3-stable semantics.*

Proof. The result follows from the observations:
1) Any G'_3-stable model M of P has the property that $M \subset Head(P)$
2) Any set of atoms $M \subset Head(P)$ models a clause of the form $A \leftarrow B^+ \wedge \neg B^-$ if and only if, it models the clause $A \leftarrow B^+ \wedge \neg(B^- \cap Head(P))$
3) The two clauses, $A \leftarrow B^+ \wedge \neg(B^- \cap M)$ and $A \leftarrow B^+ \wedge \neg(B^- \cap Head(P) \cap M)$ are the same whenever $M \subset Head(P)$

Definition 8. *The transformation RED^- deletes a rule $A \leftarrow B^+ \wedge \neg B^-$ if there is another rule $A' \leftarrow \top$ such that $A' \subset B^-$.*

Proposition 5. *For a disjunctive program P, P and $RED^-(P)$ are strongly G'_3-stable equivalent with respect to the class of disjunctive programs.*

Proof. This follows from the fact that, under the hypothesis $A' \subset B^-$, the formula $A' \rightarrow (B^+ \wedge \neg B^- \rightarrow A)$is a tautology in classical logic, for then we can derive all of the rules of P from $RED^-(P)$ and vice versa.

Definition 9. *The transformation Sub deletes a rule $A \leftarrow B^+ \wedge \neg B^-$ if there is another rule $A' \leftarrow B'^+ \wedge \neg B'^-$ such that $A' \subset A$, $B'^+ \subset B^+$ and $B'^- \subset B^-$.*

Proposition 6. *For a disjunctive program P, P and $Sub(P)$ are strongly G'_3-stable equivalent with respect to the class of disjunctive programs.*

Proof. This follows from the fact that $P \equiv_{G'_3} Sub(P)$, which in turn follows from the various implications: $B^+ \wedge \neg B^- \rightarrow B'^+ \wedge \neg B'^-$, $B'^+ \wedge \neg B'^- \rightarrow A'$ and $A' \rightarrow A$.

Definition 10. *The transformation Taut deletes a rule $A \leftarrow B^+ \wedge \neg B^-$ if $A \cap B^+ \neq \emptyset$*

Proposition 7. *For a disjunctive program P, P and $Taut(P)$ are strongly G'_3-stable equivalent with respect to the class of disjunctive programs.*

Proof. This follows from the fact that a rule $A \leftarrow B^+ \wedge \neg B^-$ for which $A \cap B^+ \neq \emptyset$ is a tautology in the G'_3-logic, and so $P \equiv_{G'_3} Taut(P)$.

Definition 11. *If P contains the clause $a \leftarrow$, and there is also a clause: $A \leftarrow B^+ \wedge \neg B^-$ such that $a \in B^+$, then the transformation Dsuc replaces it by the rule: $A \leftarrow (B^+ - \{a\}) \wedge \neg B^-$.*

It is not difficult to see that the transformation Dsuc preserves strong G'_3-stable equivalence.

Definition 12. *The transformation Failure deletes a rule: $A \leftarrow B^+ \wedge \neg B^-$, from the program P if $B^+ \cap (Head(P))^c \neq \emptyset$.*

The transformation Failure preserves G'_3-stable equivalence but it des not preserve strong G'_3-stable equivalence as the following example shows:

Example 6. Let P_1 and its transformed program P_2 according to Failure be:

$$P_1 = \{b \leftarrow, \quad a \leftarrow b, \quad d \leftarrow c\}$$

$$P_2 = \{b \leftarrow, \quad a \leftarrow b\}$$

P_1 and P_2 are G'_3-stable equivalent but they are not uniformly G'_3-stable equivalent as it can be seen by adding to P_1 and P_2 the rule: $c \leftarrow$.

4 Conclusion

The results presented in this paper generalize to the G'_3-stable semantics, and in particular, to the P-stable semantics some results dealing with equivalence and strong equivalence under the stable semantics presented in [10]. The results allow to substitute strongly equivalent programs or to add known facts to programs in order to simplify them and make easier the computation of G'_3-stable and in particular P-stable models. All the transformations presented here were studied in [10] in the context of the stable semantics, but there are some transformations presented there that we have not explored yet. Among them, the generalized principle of partial evaluation and the Dloop transformation. Those transformations offer future work to continue our research of the P-stable semantics.

References

1. W. A. Carnielli and J. Marcos. A taxonomy of C-Systems. In *Paraconsistency: The Logical Way to the Inconsistent, Proceedings of the Second World Congress on Paraconsistency (WCP 2000)*, number 228 in Lecture Notes in Pure and Applied Mathematics, pages 1–94. Marcel Dekker, Inc., 2002.

2. N. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15(4):497–510, 1974.
3. D. de Jongh and L. Hendriks. Characterization of strongly equivalent logic programs in intermediate logics. *Theory and Practice of Logic Programming*, 3(3):259–270, 2003.
4. M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. Bowen, editors, *5th Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.
5. V. Lifschitz. Foundations of logic programming. in principles of knowledge representation, pages 69-127. CSLI publications, 1996.
6. V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2:526–541, 2001.
7. J. W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, second edition, 1987.
8. M. Osorio, J. Carballido, and J. Arrazola. Logical weak completions of paraconsistent logics. *submitted to the Journal of Logic and Computation.*
9. M. Osorio and J. L. Carballido. Brief study of G'$_3$ logic. In *Unpublished paper*, 2007.
10. M. Osorio, J. A. Navarro, and J. Arrazola. Equivalence in Answer Set Programming. In A. Pettorossi, editor, *Logic Based Program Synthesis and Transformation. 11th International Workshop, LOPSTR 2001*, number 2372 in LNCS, pages 57–75, Paphos, Cyprus, Nov. 2001. Springer.
11. M. Osorio, J. A. Navarro, J. Arrazola, and V. Borja. Ground nonmonotonic modal logic S5: New results. *Journal of Logic and Computation*, 15(5):787–813, 2005.
12. M. Osorio, J. A. Navarro, J. Arrazola, and V. Borja. Logics with common weak completions. *Journal of Logic and Computation*, 16(6):867–890, 2006.
13. D. Pearce. Stable Inference as Intuitionistic Validity. *Logic Programming*, 38:79–91, 1999.

Rewards in Persuasive Group Decision Making

Ricardo Santos[1,2], Goreti Marreiros[1,3], Carlos Ramos[1,3], José Neves[4] and José Bulas-Cruz[5]

[1] GECAD – Knowledge Engineering and Decision Support Group
Porto, Portugal
{goreti,csr}@dei.isep.ipp.pt
[2] College of Management and Technology– Polytechnic of Porto
Felgueiras, Portugal
rjs@estgf.ipp.pt
[3] Institute of Engineering – Polytechnic of Porto
Porto, Portugal
{goreti,csr}@dei.isep.ipp.pt
[4] University of Minho
Braga, Portugal
jneves@di.uminho.pt
[5] University of Trás-os-Montes e Alto Douro
Vila Real, Portugal
jcruz@utad.pt

Abstract. Decision Making is one of the most important activities of the human being. Nowadays decision-making involves considering many different points of view, so decisions are commonly taken by formal or informal groups of persons. Groups exchange ideas, engage in a process of argumentation and counter-argumentation, negotiate, cooperate, collaborate and even discuss techniques and/or methodologies for problem solving. Group Decision Making is a social activity in which the discussion and results consider a combination of rational and emotional aspects. In this paper we propose a Multi-Agent model for the representation of the behaviour of individuals as members of a group involved in a decision making processes. It makes use of persuasive argumentation in the interactions between participants that makes use of a reward mechanism. The reward mechanism that we describe here consists in an algorithm to generate and evaluate problem independent rewards in order to allow participants to reach better agreements in a shorter time.

1 Introduction

Group decision making processes are omnipresent in several everyday activities. We can have more formal decision groups like committees or management teams who have to decide for instance if it is more advantageous to acquire a competitor enterprise and their technology or invest in research to develop the technology internally. Or we can have more informal groups like, for instance, a married couple who have to decide what movie to watch.

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 99–110

Received 17/06/07
Accepted 31/08/07
Final version 14/09/07

During the past 20 years, several Group Decision Support Systems (GDSS) have been developed, some dedicated to be used exclusively in decision rooms and others with features to support ubiquitous group decision meetings. The main goal of GDSS is to support a group that is responsible for decision making.

A more recent approach to Group Decision support is the use of agent based systems to support or simulate group decision processes. Multi-agent systems are very suitable for simulating the behaviour of groups of people working together [1] and, in particular, for modelling group decision making, as they cater for individual modelling, flexibility and data distribution.

The group decision-making process begins with the identification of the problem. In this stage, relevant information concerning the problem is collected. During the design phase, the group develops and discusses several possible alternative solutions for the problem and establishes the decision criteria. There are quite a few ways for generating candidate solutions, some of them are: brainstorming, reviewing the literature, conducting research and issuing a request for proposals for alternative solutions. Finally, in the choice phase, group members choose a final solution from a set of possible alternatives.

The focus of our research is in the choice phase. Argumentation can be an excellent way to justify possible choices and to convince other elements of the group that an alternative is better or worse than another. Amgoud et al. [2] states that humans often use arguments to explain past choices or to evaluate potential alternatives.

In multi-agent systems literature, argumentation is referred to as a key form of interaction among autonomous agents [3][4], particularly in non-cooperative situations and when agents have to deal with incomplete information about each other and about the environment.

In the context of negotiation, argumentation is viewed as a mechanism whereby information may be exchange. An argument is viewed as a piece of information that may allow an agent to [5]:

- Justify its negotiation decision or option; and
- Influence other agents about the quality of its proposals.

In recent years there has been a significant number of proposals for argumentation-based negotiation that use the arguments referred to above. Among the earliest proposals was Katia Sycara's Persuader System, a framework for intelligent conflict resolution and mediation.

This work led to subsequent research by Sarit Kraus et al. [4], who proposed a logical model for reaching agreements through argumentation. In this model the argumentation style is based on the psychology of persuasion, using threats, rewards, appeals to precedents, appeals to self-interest and appeals to prevailing practices. Agents are characterized over several attributes: whether they are less, knowledgeable, cooperative, etc. The selection of the arguments is based on a pre-defined order.

Sierra et al. [6] proposed a generic framework for argumentation-based negotiation, in which agents exchange proposals and counter-proposals justified by arguments. The presented argumentation protocol is based on a finite state machine where exchanged arguments are persuasive: threats, reward and appeal. The evaluation of the arguments received by an agent is in part based on the Dung argumentation theory and its notion of argument acceptability. In this work it is not defined how agents

select arguments to send to other agents, there being no pre-defined order as in the work of Sarit Kraus [4].

In the literature it is possible to see references to threats and promises as the most used arguments in human discussions.

This paper is organized as follows: Section 2 presents the multi-agent model of our system and is also presented the architecture of participant agents [7] describing its main components and interactions. Section 3 describes with more detail the argumentation system responsible for the arguments generation. Section 4 presents the reward mechanism. Section 5 presents some conclusions and directions for future work.

2 Multi-Agent Model and Participant Agent

Agent Based simulation is considered an important tool in a broad range of areas e.g. individual decision making (what if scenarios), e-commerce (to simulate the buyers and sellers behaviour), crisis situations (e.g. simulate fire combat), traffic simulation, military training, entertainment (e.g. movies).

According to the architecture that we are proposing we intend to give support to decision makers in both of the aspects identified by Zachary and Ryder [8], namely supporting them in a specific decision situation and giving them training facilities in order to acquire competencies and knowledge to be used in a real decision group meeting. We maintain that agent based simulation can be used with success in both tasks. As stated in the introduction multi-agent systems seem to be quite suitable for simulating the behaviour of groups of people working together [1][9].

Each participant of the group decision making process is associated with a set of agents for interaction with other participants. The community should be persistent because it is necessary to have information about previous group decision making processes, particularly the credibility, reputation and past behaviors of other participants [10].

There are three different types of agents in our model: Facilitator agent, Assistant agent and the Participant agent.

The Facilitator agent helps the responsible for the meeting in its organization (e.g. decision problem and alternatives definition). During the meeting, the Facilitator agent will coordinate all the processes and, at the end, will report the results of the meeting to the participant responsible for it.

The Assistant agent works as an assistant to the participant of the meeting presenting all the updated information of the meeting. This agent acts like a bridge between the participant (user) and the participant agent.

The participant agent has a very important role in the group decision support system, for that reason we will present the architecture and a detailed view of all the component parts.

The architecture is divided in three layers: the knowledge layer, the communication layer and the reasoning layer (Fig.1).

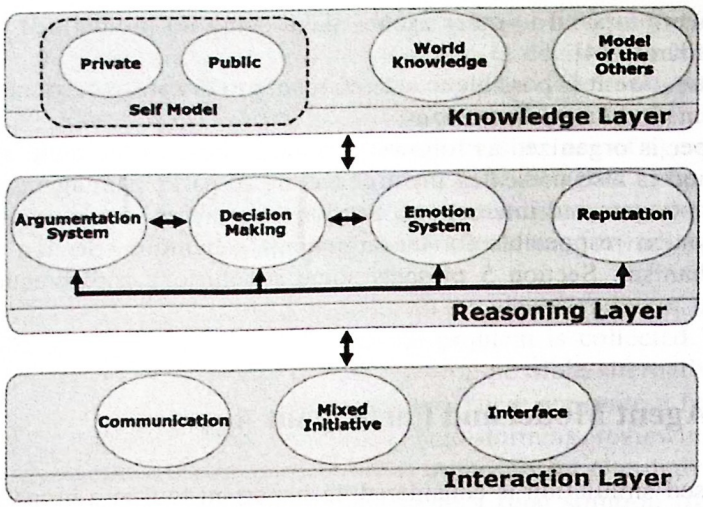


Fig. 1. Participant Agent Architecture

In the knowledge layer the agent has information about the environment in which it is situated, about the public profile of the other participant agents that compose the decision meeting group, and regarding its own preferences and goals (its own public and private profile). The information in the knowledge layer is dotted of uncertainty [8] and will be accurate for the duration of the interactions done by the agent.

The interaction layer is responsible for communication with other agents, the interface with the user of the group decision making simulator and the mixed initiative interaction between participant and agents.

The reasoning layer contains three major components:

- The argumentative system – that is responsible for the generation of arguments. This system will be described in more detailed in section 3.
- The decision making module – will support agents in the choice of the preferred alternative and will classify all the set of alternatives in three classes: preferred, indifferent and inadmissible;
- The emotional system [11] – will generate emotions and moods, affecting the choice of the arguments to send to the others participants, the evaluation of the received arguments and the final decision.
- The reputation module – this module tries to support the user in the definition of the level of trust given to the Participant agent in the delegation of actions.

3 Argumentation System

Arguments may be classified according to type. Here we assume that the following six types of argument have persuasive force in human negotiations [12][13]: threats; promise of a future reward and appeals; appeal to past reward; appeal to counter-example; appeal to prevailing practice; and appeal to self interest. These are the ar-

guments that agents will use to persuade each other. This selection of arguments is compatible with the power relations identified in the political model: reward, coercive, referent, and legitimate [14].

This component will generate persuasive arguments based on the information that exists in the participant's agent knowledge base [15].

3.1 Argumentation Protocol

During a group decision meeting, Participant agents may exchange the following locutions: request, refuse, accept, request with argument.

Request (AgP_i, AgP_j, α, *arg*) - in this case agent AgP_i is asking agent AgP_j to perform action α, the parameter *arg* may be void and in that case it is a request without argument or may have one of the arguments specified in the end of this section.

Accept (AgP_j, AgP_i, α) - in this case agent AgP_j is telling agent AgP_i that it accepts its request to perform α.

Refuse (AgP_j, AgP_i, α) - in this case agent AgP_j is telling agent AgP_i that it cannot accept its request to perform α.

The purpose of the agent participant its to replace the user when he is not available. For example, in Fig. 2, it is possible to see the argumentation protocol for two agents. However, note that one of the participants is not available at the moment leaving all the actions to the participant agent (AgP_2). Note that his is the simplest scenario, because in reality, group decision making involves more than two agents and, at the same time AgP_1 is trying to persuade AgP_2 that agent may be involved in other persuasion dialogues with other group members.

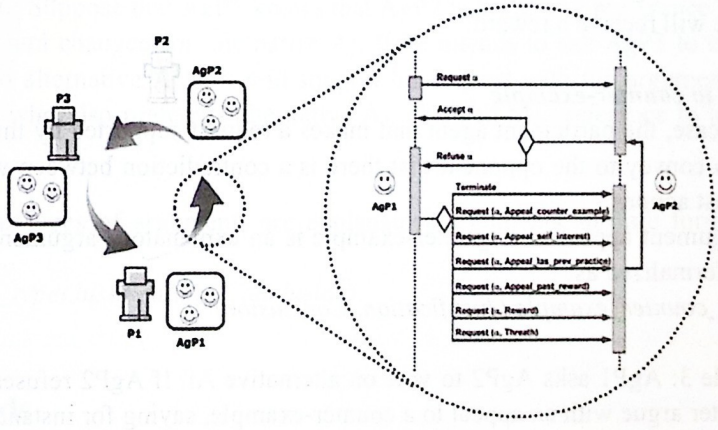


Fig. 2. Argumentation protocol

The autonomy of the participant agent is connected with the trust that the real participant has in the agent. As the agent exchanges locutions with other participants the user can approve or reject the locutions made by the participant agent there are no

definitive locutions made by the agent. In that case trust level can be increased or decreased.

3.2 Generation of Arguments

In this section we describe and specify the persuasive arguments that our agents may change. For each argument a short example is given.

Threats

As previously stated, threats are very common in human negotiation, and they can assume two distinct forms: you should perform *action A* otherwise I will perform *action B*; and you should **not** perform *action A* otherwise I will perform *action B*.

In our model this type of arguments may be formalized as a triplet:

Threat(Justification, Conclusion, Threatened_goal)

Example 1: AgP1 asks AgP2 to vote on alternative Ai with the argument that if he refuses he will vote on alternative Aj that he believes is unacceptable for AgP2.

Promises

In this case, the participant that has a request refused can offer a reward for the acceptance of the request. The reward can be claimed by the opponent in the future. This mechanism of rewards will be described in more detail in section 4.

In our model this type of arguments may be formalized as a triplet:

Promise(Justification, Conclusion, Promised_goal)

Example 2: AgP1 asks AgP2 to vote on alternative Ai with the argument that if he accepts he will receive a reward.

Appeal to counter-example

In this case, the participant agent that makes a request supported by this argument, expects to convey to the opponent that there is a contradiction between what he says and his past actions.

The argument appeal to a counter-example is an explanatory argument and in our model is formalized as:

Appeal_counter_example (Justification, Conclusion)

Example 3: AgP1 asks AgP2 to vote on alternative Ai. If AgP2 refuses then AgP1 may counter argue with an appeal to a counter-example, saying for instance that in the past it preferred alternative Ak and changed to Al, so why not to do the same now?

Appeal to self-interest

In this case the participant agent that makes a request supported by this argument expects to convince his interlocutor that making action *A* is in his best interest.

The argument appeal to self-interest is an explanatory argument and in our model is

formalized as:

Appeal_self_interest (Justification, Conclusion)

Example 4: Suppose that AgP1 asks AgP2 not to vote for alternative Ai, supported by the argument that AgP3 voted for the Ai alternative and AgP2 does not like AgP3.

Appeal to past reward

In this case the participant agent that sends such an argument expects his interlocutor to perform an action based on a past promise.

In our model the appeals to past rewards are formalized as:

Appeal_past_reward (Promise(Justification, Conclusion, Promised_goal), Conclusion)

Example 5: If at some point in the past agent AgP1 sent a request to AgP2 to vote on a specific alternative with the promise of a reward that he will stay in a debt of gratitude, and if AgP2 accepted it, in a future decision AgP2 can send a request supported by this argument.

Appeal to prevailing practice

In this case, a participant agent believes that the opponent agent will refuse to perform a requested action since it contradicts one of its own goals. For that reason the participant agent sends a request with a counter-example from a third agent's actions or from past actions of the opponent.

In our model the appeal to prevailing practice is formalized as:

Appeal_prevailing_practice (Justification, Conclusion)

Example 6: Suppose that AgP1 knows that AgP2 had a strong preference for the alternative Ai and changed for alternative Aj. If he intends to ask AgP3 to change his preference to alternative Aj he could support his request with the argument that another agent, who also preferred alternative Ai, changed his preference to alternative Aj.

The last four types of arguments are explanatory arguments and are formalized as tuple:

Argument_type(Justification, Conclusion).

4 Rewards

Usually the participants of a meeting do not all share the same objectives, so it is usually difficult to reach a consensus. Rewards are an extremely important issue in reaching an agreement on the selection of an alternative in a group decision meeting. In the negotiation process participants send requests to others that can be accepted or rejected depending on their personal profile and objectives. In order to obtain better results via unanimous consensus we think it is necessary to have a method to define

rewards independent of a specific problem that can be used to justify locutions in future meetings. Depending on the outgoing of the meeting, Participants can offer rewards, evaluate received proposals, and, in the future, in other requests or meetings claim the promised rewards.
Our approach to the rewards mechanism was developed independently of any problem domain and it is divided in two parts: the process of the reward offering; and the process of the reward evaluation. To explain our approach we will use a simple scenario involving two Participant agents: AgP_1 and AgP_2. AgP_1 is the agent participant that in the middle of a meeting more precisely in the negotiation process decides to send a request with a promise of a future reward to AgP_2. AgP_2 is the participant agent who should evaluate that request.

4.1 Process of Reward Offering

The offer of the reward is the responsibility of AgP_1, who sends a request in order to secure an acceptance of the alternative in question. If the Participant agent AgP_2 accepts the reward, in the future he can claim the reward related to this alternative, which may or may not be provided. In more detail, the next algorithm outlines the main steps of the Participant agent AgP_1.

1. Examine the request refused.
2. Calculate a value to formulate the reward.
3. Offer the reward.

Step 1: Examine the request refused

In this first step AgP_1 will examine the public profile of AgP_2, in order to find the reason for the rejection of the request. The refused request has to be analyzed in order to calculate the reward in the next step. From this analysis we can predict several possible results:

- All arguments are defined and in favor of the criteria of the alternative in the public knowledge; the request was refused by a direct order coming from the participant user;
- All arguments are defined in the public knowledge, some against and others in favor of the alternative; the request was refused because there are arguments against the alternative;
- Some arguments are defined as part of the public knowledge and others not. The other missing arguments could be in the private knowledge or could even be not defined;
- There are no arguments defined in the public knowledge. The arguments could be in the private knowledge or not defined.

The first possibility presented is an exception to the reward calculus because all of the criteria for the requested alternative are proved in the public knowledge of the participant. No reward can be defined because all arguments are in favor of the alternative and the request was refused intentionally by the participant user. This action brings us to a problem that is the possibility of the participant user to refuse a pre-

ferred request intentionally in order to collect a future reward. There are two distinct ways of solving this problem, one for the Participant agent and the other for the participant user:

- The participant agent may move to a threat argument of the kind "*If you do not accept this request I will refuse your request(s)*";
- The participant user may argue with the other participant.

The other possibilities are solved through the calculus of a maximum reward to be offered to the Participant agent AgP_2, explained in the next step.

Step 2: Calculate a value to formulate the reward

To determine the value of the reward we use a utility function to predict a fair value to offer to AgP_2. This function uses the self-knowledge of AgP_1 to determine the value of the reward, so what the reward will not be over evaluated. This issue has some importance by the fact that in the future AgP_1 will be in debt to AgP_2. The utility function its defined afterwards:

$$URewards(Argument_x) = \frac{WArgument_x}{\sum_{i=1..n} WArgument_i}$$

Where $Argument_x$ is an argument that proves a criterion of an Alternative and $WArgument_x$ is the importance that the participant user gives to the argument in question. For the following alterantive: $Alternative_1(Price=10)$ a possible argument can be $(Price < 15)$ and the Wargument $(60\ [0\text{-}100])$.

As explained previously this function must calculate a reward in several scenarios: when we know what arguments are contrary to the alternative criteria or when these are unknown. To illustrate the use of the reward utility function we will present an algorithm that estimates the maximum reward value. Consider the following assumptions:

- $Request(AgP_1,AgP_2,Alternative_x)$ as a request made by AgP_1 to AgP_2 to accept the alternative$_x$ that has been refused;
- $Alternative_x(Citeria_1,Criteria_2,\ldots,Criteria_n)$ as the alternative$_x$ with a set of criteria proved by a set of arguments $Argument_1,Argument_2,\ldots,Argument_n$ in the public knowledge of the participant.
- $PublicKnowledge_{AgP2}$ as the public knowledge of AgP_2;
- $UPrefAlternative_x$ is the utility of the preferred alternative$_x$ being in the range $[0..1]$.

Begin

Foreach $Argument_i \in Alternative_x$ ***do***

If $((Argument_i \notin PublicKnowledge_{AgP2}) \ || \ (\neg Argument_i \in PublicKnowledge_{AgP2}))$ ***then***

$$Max_Reward \leftarrow Max_Reward + \frac{WArgument_i}{\sum_{j=1..n} WArgument_j}$$

End if

End for
$Max_Reward \leftarrow Max_Reward * UPrefAlternative_x$
End

Step 3: Offer the reward

The step of the offering reward consists in an iterative algorithm that sends promises of rewards containing the maximum reward value calculated in the previous step as a limit in order to make incremental reward offers to the opponent. The purpose of this algorithm its to offer the minimum possible reward to a selected opponent because when there is no information about some criteria of an alternative it is not possible to know which arguments are in favor, neutral or against. The algorithm of the reward offers is presented next. Consider:

- *NOfUnknownArgs* – number of unknown arguments for the alternative criteria;
- $Alternative_x$ – the $Alternative_x$ is the alternative for which acceptance is requested;
- *Justification* – arguments that justify the promise. This parameter can be empty;
- *Tolerance%* - percentage value of the reward that can be increased;
- *RequestAccepted* – can be true or false. If true the request sent was accepted if false the request was rejected.

Begin
$i \leftarrow 0$
Do
$Request(AgP_1, AgP_2,$
$Promisse(Justification, Alternative_x, \frac{Reward}{NOfUnkownArgs - i++}))$
while$(i<NOfUnknownArgs)$ && $(\neg RequestAccepted))$
if $((\neg RequestAccepted)$ ***then***
$Request(AgP_1, AgP_2,$
$Promisse(Justification, Alternative_x, Reward_x * (1 + tolerance\%)))$
end if
End

4.2 Process Of Reward Evaluation

The reward evaluation is the responsibility of AgP_2 who must analyze the requests received to see if the proposition is satisfactory. Bellow there is an example definition of a request with a reward proposition:

$$Request(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward))$$

After the receipt of this request, the following algorithm is executed to evaluate it, so that it may be accepted or rejected. In case of a rejection AgP_1 can propose another reward.

Begin
Foreach $Argument_i \in Alternative_x$ ***do***
If $((Argument_i \notin PublicKnowledge_{AgP2}) \mid\mid (\neg Argument_i \in PublicKnowledge_{AgP2}))$ ***then***

$$Pretended_Reward \leftarrow Pretended_Reward + \frac{WArgument_i}{\sum_{j=1..n} WArgument_j}$$

End if
End for
$Pretended_Reward \leftarrow Pretended_Reward * (1 - WPrefAlternative_x^{-1})$
If $Reward \geq Pretended_Reward$ ***then***
$Accept(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward))$
Else
$Refuse(AgP_1, AgP_2, Promisse(Justification, Alternative_x, Reward))$
End if
End

Where *WPrefAlternative* is a value between [0..1] that represents the interest of the participant in that particular alternative.

5 Conclusion

In this work an agent-based architecture is proposed to support a group decision support system that considers persuasive argumentation between participants. In the persuasive argumentation it is possible to offer rewards to other participants of the meeting that can be claimed in the future. The reward mechanism defined in this paper is divided in two parts: the process of reward offering and the process of reward evaluation. In terms of the reward offering process we have created two algorithms, one to evaluate the maximum reward possible to offer to a specific opponent and another one to make the offer of the reward. In terms of the reward evaluation process we have also created and algorithm to evaluate the offered reward in order to verify if it is a fair and interesting offer according to the profile of the participant.

Future work includes the introduction of the reputation of the meeting participants in the rewards mechanism.

References

1. G. Marreiros, C. Ramos, J. Neves. "Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision", Lecture Notes in Computer Science, 3823 pp. 41-50, 2005.
2. Amgoud, L.; J. Bonnefon and H. Prade, (2005). An Argumentation-Based Approach to Multiple Criteria Decision. Lecture Notes in Computer Science, 3571, pp. 269 – 280.
3. Parsons, S. ; Sierra, C.; Jennings, N.R. – Agents that reason and negotiate by arguing. Journal of Logic and Computation, vol. 8 n. 3 pp 261-292, 1998.
4. Kraus, S.; K Sycara and A. Evenchick (1998). Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 104(1-2) pp 1-69.
5. Jennings, N.; S. Parsons; P. Noriega and C. Sierra (1998). On Argumentation-Based Negotiation. Proceedings of the International Workshop on Multi-Agent Systems, USA.
6. Sierra, C.; Jennings, N.R.; Noriega,P.; Parsons, S. (1998). A framework for argumentation-based negotiation. Intelligent agents IV LNAI, Vol.1365 pp. 177-192.
7. G. Marreiros, R. Santos, C. Ramos, J. Neves, P. Novais, J. Machado, J. Bulas-Cruz, "Ambient Intelligence in Emotion Based Ubiquitous Decision Making", Proc. Artificial Intelligence Techniques for Ambient Intelligence, IJCAI'07 – Twentieth International Joint Conference on Artificial Intelligence. Hyderabad, India, 2007.
8. W. Zachary and J. Ryder, "Decision Support Systems: Integrating Decision Aiding and Decision Training". In: Handbook Of Human-Computer Interaction, pp. 1235-1258. The Netherlands: Elsevier Science, 1997.
9. P. Davidsson, "Multi agent based simulation: beyond social simulation", Proc. of the Second international Workshop on Multi-Agent Based Simulation. S. Moss and P. Davidsson, Eds. Springer-Verlag NJ, pp. 97-107, 2001.
10. F. Andrade, J. Neves, P. Novais, J. Machado, A. Abelha, "Legal Security and Credibility in Agent Based Virtual Enterprises, in Collaborative Networks and Their Breeding Environments", Springer-Verlag, pp 501-512, 2005.
11. R. Santos, G. Marreiros, C. Ramos, J. Neves, J. Bulas-Cruz. "Multi-agent Approach for Ubiquitous Group Decision Support Involving Emotions". Lectures Notes for Computer Science, 4159, pp. 1174 – 1185, 2006.
12. D. O'Keefe, "Persuasion: Theory and Research", SAGE Publications, 1990.
13. D. Pruitt.,"Negotiation Behavior", Academic Press, New York, N.Y., 1981.
14. G. Salancik, J. Pfeffer, "Who Gets Power – And how they hold on to it- A Strategic Contingency Model of Power. Organizational Dynamics, Vol. 5, pp. 3-21, 1977.
15. G. Marreiros, C. Ramos e J. Neves. "Multi-Agent Approach to Group Decision Making through Persuasive Argumentation". Proc. 6th International Conference on Argumentation (ISSA'06), Amesterdam, 2006.

Representation and Reuse of Design Knowledge and Design Rationale for Design Decision-Making

Jie Hu[1], Yinghong Peng[1],
Guangleng Xiong[2]

[1] School of Mechanical Engineering, Shanghai Jiao Tong University,
Shanghai, P.R. China
hujie@sjtu.edu.cn
[2] Department of Automation, Tsinghua University,
Beijing, P.R. China
glxiong@cims.tsinghua.edu.cn

Abstract. Design rationale comprises information from two aspects: information about decision making and information about the artifact itself. This paper presents an approach of representation and reuse of design rationale for design decision-making. The relations between design description and design decision are analyzed and the conditions for consistence are given. The ontologies for design rationale representation are described. The constraints of consistence are deducted based on the conditions and the design rationale model. A system for presentation and reuse of design knowledge and design rationale is developed. The method described in this paper was used to develop a system, which is applied to presentation and reuse of design knowledge and design rationale for a railway vehicle bogie design.

Keywords: Knowledge representation, Design rationale, Ontologies, Consistence, Decision-Making.

1 Introduction

Knowledge about the rationale for a design—how and why a device is designed as it is—can be valuable, but is difficult to capture in reusable form. A design rationale is an explanation of how and why an artifact, or some portion of it, is designed the way it is. A design rationale is a description of the reasoning justifying the resulting design—how structures achieve functions, why particular structures are chosen over alternatives, and what behavior is expected under what operating conditions. In short, a design rationale explains the "why" of a design by describing what the artifact is, what it is supposed to do, and how it got to be designed that way [1].

Design rationale represents design knowledge that is used by people—possibly assisted by interactive computer aids—for engineering tasks including manufacturing, verification, diagnosis, and redesign. In current engineering practice, this knowledge is often lost because designers forget it, leave the project, or are inaccessible in a large organization.

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 111–119

Received 16/06/07
Accepted 31/08/07
Final version 21/09/07

The design documentation at present focuses on the data about "how to implement the design artifact", while the information about "why the product is designed as it is" is ignored. These information is believed to be as important as implementation data. It helps to support design reasoning, to facilitate design communication and to further design knowledge accumulation and development so that the quality and efficiency of product design are improved and the product competence of enterprises are enhanced. Design rationale (DR) research is exploring the representation, capture and uses of these information. Explicitly represented rationale can help individual designers clarify their thinking about a design [2-6]. Perhaps more importantly, rationale capture can support design by concurrent engineering (CE) teams over time. The reasoning behind decisions becomes available for all team members to critique and augment. Most existing rationale capture approaches support only individual users and are thus not suited to team contexts.

Representing design rationale requires that one explicitly document the reasoning and argumentation in design [7]. Szykman *et al.* [8] presented two fundamental representations of design rationale. First is the notion of design rationale as the recording of the design intent of an artifact. For example, in traditional mechanical design, rationale might include a functional description, geometric or assembly constraints and performance criteriaThe second view is of design rationale as a record of the design process, the communications among agents, the decision-making that occurs, as well as the decision-making process. This view of rationale has been often studied in the software design and computer-supported collaborative work communities, where one goal is to support organizational intelligence and group decision making.

This paper is to solve the problem of "how to represent and reuse design rationale in collaborative product design" based on a combination of current design rationale research and engineering practices in manufacturing enterprises. The relations between design description and design decision are analyzed and the conditions for consistence are given. A system is developed and applied to presentation and reuse of design knowledge and design rationale for a railway vehicle bogie design.

2 Representation of Design Knowledge and Design Rationale

Multidisciplinary collaborative design is a methodology for the design of complex coupled systems, in which the synergistic effects of coupling between various interacting disciplines are explored. These subsystems are usually geographically distributed and implemented in different computers to support complex design projects carried out in multidisciplinary teams contexts.

Multidisciplinary designers are no longer merely exchanging geometric data, but more general knowledge about design and the product development process, including specifications, design rules, design constraints, and design rationale. Furthermore, this exchange of knowledge often crosses corporate boundaries. As design becomes increasingly knowledge-intensive and collaborative, the need for computational frameworks to enable engineering product development, by effectively supporting the formal representation, capture, retrieval and reuse of product knowledge, becomes more critical.

Design rationale comprises information from two aspects: information about decision making (issues, alternatives, criteria, objectives) and information about the artifact itself (requirements, function, structure, behavior). To support the four kinds of computer services, we build an integrated information model of the two aspects. The model is illustrated in Fig. 1. The concepts and relations between them are explained as follows.

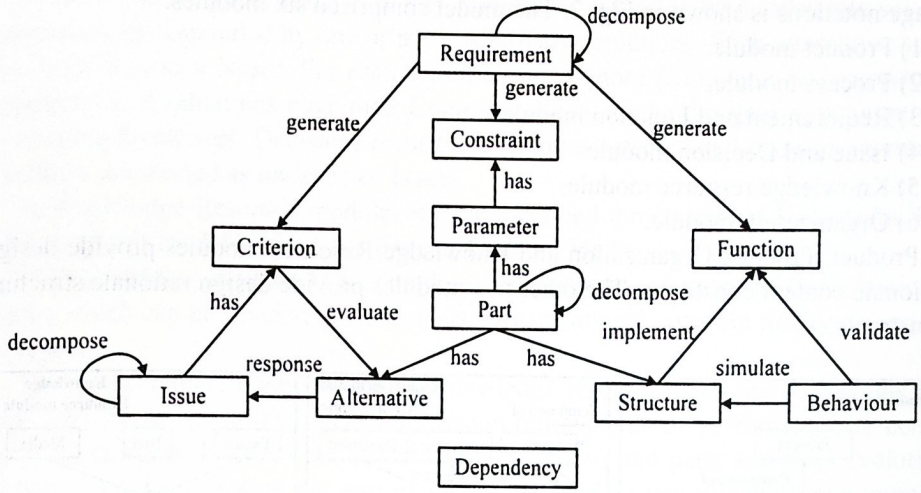


Fig.1. The design rationale model.

Requirement: The specification describing the function, structure and behavior of the design. It can be divided into criterion and constraint.

Criterion: The expectation for measurable product properties, such as cost, manufacturability, safety, etc.

Issue: A decision problem. It involves a choice among a set of alternatives, relevant criteria, and information used to compute the utilities of the alternatives in terms of criteria.

Alternative: The description of design or some part of it. Decisions are choices among alternatives.

Constraint: An expression specifying a limit on the range of values of one or more parameters.

Parameter: A variable describing some property of design artifact. Parameters can describe the function, constraint and behavior of the design.

Part: Elements composing of product design. It can be components, assemblies, or subsystems.

Function: Expected behavior of design artifact in some operation environments.

Structure: The physical and/or logical composition of an artifact, typically in terms of composition of parts, and connection topologies. Structure description is the basis for realizing the artifact and is depicted with schematics or drawings.

Behavior: States and changes of the artifact in some operation environment. Behavior descriptions are relations over quantities of parameters if states are described by these parameters.

Dependency: Determination relations between design element (such as parameters, requirements, constraints and issues). The relations are unidirectional. Dependency management is necessary because design reuse must involve design modifications.

These concepts and relations between them form the core of design rationale model.

The ontologies to describe design rationale model using Unified Modeling Language notations is shown in Fig. 2. The model comprised six modules:

1) Product module,
2) Process module,
3) Requirement and Function module,
4) Issue and Decision module,
5) Knowledge resource module,
6) Organization module.

Product, Process, Organization and Knowledge Resource modules provide design rationale context constructs. The other two modules provide design rationale structure constructs.

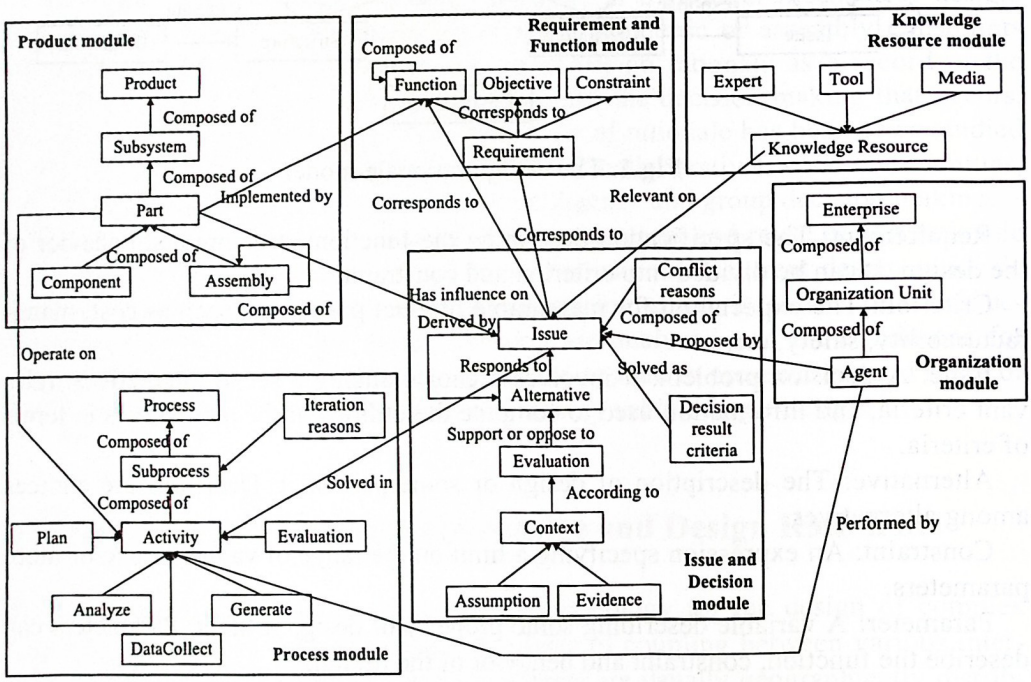


Fig. 2. The ontologies to describe design rationale model.

The objects and relations in the model are explained concisely as follows:

1) Product module. In this module, Products are composed of subsystems. Subsystems are composed of Parts. There are two kinds of Parts: Components are indecomposable and Assemblies are composed of Components and subassemblies.

2) Process module. In this module, Processes are composed of Subprocesses. Subprocesses are composed of Activities and sub-Subprocesses. Activities can not be divided further. We define five kind of Activity: Plan, Analyze, Data Collect, Generate and Evaluate. Iteration is modeled as a kind of Subprocess.

3) Requirement and Function module. In this module, Requirements and Functions can be decomposed to more specific Requirements and Functions. Requirements are distinguished as initial requirements and derived requirements. Initial requirements are explicit statements of customers. Derived requirements are derived from initial requirements during the design process. Functional demands, Objectives, and Constraints can be extracted from Requirements.

4) Issue and Decision module. In this module, An Issue can be divided into several sub-Issues, or responded by one or more Alternative solutions. Each alternative solution may raise new issues. For each solution, one or more Evaluation may support or oppose to it. Evaluations have their Contexts, which can be designer's Assumptions or existing Evidences. Decisions record the solutions selected and decision rationale. Conflicts are treated as unexpected Issues.

5) Knowledge Resource module. We have defined three kinds of Knowledge Resources in this module, which are Expert, Tool and Media.

6) Organization module. In this module, Enterprise is composed of Organizational Units, which can be divided into sub-units. Organizational units are finally composed of Agents.

The parts, requirements, functions, knowledge resources, activity and agents are explicitly modeled, so intelligent information retrieval can be performed. The corresponded relations between the requirements, functions and parts and their evolution during the design process are also explicitly modeled so that effective requirements management can be performed. The multiple views to present design rationale can also be constructed on the information model.

3 Consistent Analysis of Design Rationale for Design Decision-Making

For reuse of design ration, the consistent of design rationale should be analyzed. The consistent analysis algorithm of design rationale can be described as:

```
Procedure Consistence-Checking(Rationale dr)
Begin
/*Decision behaviour checking*/
  /*Constraints for feasibility*/
  For i From 1 to M=total number of alternatives Do
    IF(selected(Ai) And totalEvaluation(Ai)<0) Then
      Print("feasibility constraints violated!",i);
      Break;
  /*Constraints for optimality*/
  For i From 1 to M=total number of issues Do
    For j From 1 To N=total number of alternatives of
                    Issue i Do
      If (selected(Aj) And totalEvaluation(Aj)!
                          =maxTotalEvaluation(Ii)) Then
      Print("optimality constraints violated!",i);
      Break;
/*Decision content checking*/
  /*Constraints for requirements*/
```

```
For i From 1 to M=total number of requirements Do
  IF(decomposed(Ri) Then
    For j From 1 To N=total number of keywords of Ri Do
      If(notContainedIn(Kj,keywords
        (subRequirements(Ri)))) Then
      Print("requirement constraints violated!",i);
    Break;
  IF(generated(Ri) Then
    For j From 1 To N=total number of keywords of Ri Do
      if(notContainedIn(Kj,keywords
        (generatedIssues(Ri)))) Then
      Print("requirement constraints violated!",i);
      Break;
/*Constraints for issues*/
For i From 1 to M=total number of issues Do
  IF(decomposed(Ii) Then
    For j From 1 To N=total number of keywords of Ii Do
      If(notContainedIn(Kj,keywords(subIssues(Ri))))
      Then
      Print("issue constraints violated!",i);
      Break;
  IF(responded(Ii) Then
    For j From 1 To N=total number of keywords of Ii Do
      If(notContainedIn(Kj,keywords
        (respondedAlternatives(Ii)))) Then
      Print("issue constraints violated!",i);
    Break;
      If(notContainedIn(Kj,keywords
        (relatedCriteria(Ii)))) Then
      Print("issue constraints violated!",i);
      Break;
/*Constraints for alternatives*/
For i From 1 to M=total number of alternatives Do
  For h From 1 To L=total number of derived Issues of Ai
  Do flag=0;
    For j From 1 To N=total number of keywords of Ii Do
      If(containedIn(Kj,keywords(derivedIssues(Ai,h))))
      Then flag=1;
      Break;
      If(flag==0) Then
      Print("alternative constraints violated!",i)
      Break;
For i From 1 to M=total number of alternatives Do
  flag=0;
  For j From 1 To N=total number of keywords of Ai Do
    If(containedIn(Kj,keywords(relatedArgument(Ai))))
    Then flag=1;
    Break;
    If(flag==0) Then
    Print("alternative constraints violated!",i)
    Break;
/*Constraints for criteria*/
For i From 1 to M=total number of criteria Do
  flag=0;
```

```
    For j From 1 To N=total number of keywords of C_i Do
      If(containedIn(K_j,keywords(relatedArguments(C_i))))
      Then flag=1;
      Break;
      If(flag==0) Then
      Print("criterion constraints violated!",i)
      Break;
End
```

4 System and Application

We develop a system for presentation and reuse of design rationale. The functionality of the system includes three aspects: design rationale capture, design rationale browsing and design rationale query.

1) The capturing of design rationale should be integrated firmly with design process so that designers could record and index design rationale information while performing design tasks. The tools of requirements management are integrated into the design process support system. The structure of design rationale information is formed during capturing with the support tools. The contexts of design rationale are generated from the design activities. The contents of design rationale are indexed according to domain ontologies by the designers interactively.

2) Design rationale is browsed through multiple views. Each view presents some aspects of the contents, structure and contexts of the design rationale information so that the access to design rationale information resources can be guided.

3) Design rationale can be queried by its properties, contexts and contents. The information can be browsed through proper views after the instances are found by query.

The architecture of the system has three levels:

1) Application level. In this level, the tools of design rationale capturing, browsing and querying are integrated with design process support system. Services are actively provided according to specific design task.

2) Description level. In this level, design rationale information is described based on ontologies. The contexts of design rationale information are described using enterprise ontology, including departments, processes, and activities, etc. The basic properties of design rationale information are described using information ontology, including media, abstract level, etc. The contents of design rationale information are described using domain ontology to specify domain keywords. These descriptions are the basis for intelligent information retrieval.

3) Object level. This level is comprised of several kinds of design rationale information instances. Instance classes include issues, parts, documents, requirements, etc.

Following is an example to illustrate how to application the methods and system to design of a railway vehicle bogie. The bogie in the example is shown in Fig. 3.

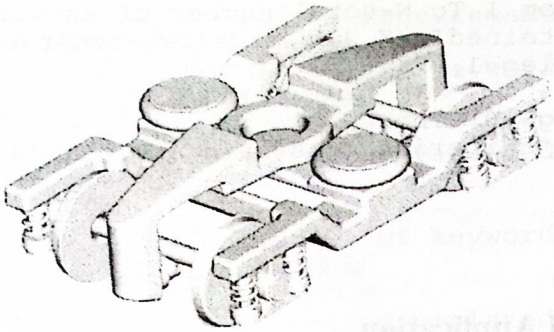


Fig.3. An example of bogie design.

In this example, issues, alternatives, criteria and related document objects are indexed using enterprise ontology, information ontology and domain ontology so that intelligent information retrieval can be performed. The contexts of design rationale information are described using enterprise ontology, including departments, processes, and activities, etc. The basic properties of design rationale information are described using information ontology, including media, abstract level, etc. The contents of design history information are described using domain ontology to specify domain keywords. The design rationale information instance is described as follows.

```
Class:           Issue
Name:            Spring selection
Media:           Text
Abstract level:  Qualitative
Requirement:     Stability, safety
Function:        Load allocation, concussion alleviation,
                 moving stability assurance
Product:         Australian food car
Subsystem:       Bogie
Part:            Spring
Process:         Australian food car development
Activity:        Bogie spring design
Department:      Design department
Designer:        J Zhang
Keywords:        Freight car bogie, spring, elasticity,
                 concussion alleviation
Sub-issue:       None
Super-issue:     Freight car bogies design
Solutions:       Round spring
```

Decision support by multi-attribute utility theory is provided to avoid personal bias. For each decision issue, there are a set of criteria and an objective function to let the designer to select between alternatives. Criteria include aspects such as cost, safety, manufacturability, etc. Each alternative is evaluated against the criteria and given an evaluation point based on the objective function, which makes a trade-off between criteria. Different criterion will have different power in the objective function and the power may change due to different requirements. Facts and assumptions which lead to the evaluation point are also recorded.

5 Conclusions

This paper presents a method of representation, reuse and consistent analysis of design knowledge and design rationale. By describing design rationale information with homogeneous ontologies, heterogeneous engineering information sources can be accessed uniformly. This ontology-based approach overcomes knowledge acquisition bottlenecks of design rationale system construction. The consistence is deducted based on the conditions and the design rationale model. The scheme of representation and consistent analysis is the basis for design decision-making.

Acknowledgement

This paper is supported by the National Natural Science Foundation of China (Nos. 50575142 and 60304015), and the Shanghai Committee of Science and Technology (Nos. 055107048 and 04ZR14081)

References

1. Gruber, T. R. and Russell, D. M.: Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use. Technical Report KSL 90-45, Stanford University. (1991) 1-40
2. Lee, J. and Lai, K.Y.: What's In Design Rationale?. Human-Computer Interaction 6 (1991) 251–280
3. MacLean, A., Young, R., Bellotti, V., and Moran, T.: Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction 6 (1991) 201–250
4. Klein, M. Supporting Conflict Resolution in Cooperative Design Systems. IEEE Systems Man and Cybernetics 21 (1991) 1379-1390
5. Shum, S. A. Cognitive Analysis of Design Rationale Representation. PhD dissertation. University of York (1991)
6. Lee, J. Design Rationale System: Understanding the Issues. IEEE Expert 12 (1997) 78–85
7. MacLean, A., Young, R., Belloti, V., and Moran, T.: Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction 6 (1991) 201–250
8. Szykman, S., Sriram, R. D. and Regli, W. C.: The Role of Knowledge in Next-generation Product Development Systems. Journal of Computing and Information Science in Engineering 1 (2001) 3–11

Modeling Ontology-Based Task Knowledge for PMBOK: A Three-Layer Method

Kwoting Fang[1], Chingwei Chang[2] ,Yenping Chi[2]

[1]National Yunlin University of Science & Technology,Yunlin, 123 University Road, Section 3, Touliou, Yunlin, Taiwan
fangkt@yuntech.edu.tw
[2]National Chengchi University,NO.64,Sec.2,ZhiNan Rd.,Wenshan District,Taipei City 11605,Taiwan
{channing,ypchi}@mis.nccu.edu.tw

Abstract. The main purpose of this study was twofold. First, it adopted the theory of task ontology to build up a two-level mediating representation for a task analysis and the task of "Risk Management" served as an example. Five phases, task analysis, task ontology, IDEF0 model, Petri net model, and PNML, are displayed for domain experts and can further transfer for computer to generate code. Second, from modeling standpoint, a methodology, called TTIPP, was provided to systemically analyze the process of the task and subtask, in terms of the inputs, outputs, mechanisms, controls, using IDEF0 and Petri net. It is hope that the results of task knowledge on risk management for project management domain, represented by ontology, can give valuable and reusable problem solving process knowledge for person with the similar goals.

Keywords: Task Ontology, IDEF0/Petri net, Project Managemen

1 Introduction

During the past few decades, rapid technology advances has witnessed the rise of a service sector, which provides knowledge-personnel to work on IT or ICT projects. Currently, technology play the vital role in every branch of the corporate value creation system, that IT projects have poorly defined goals but well-defined methods further underlines the need to investigate how these complex, high-cost, high-risk solution-based projects can be managed effectively and efficiently. Therefore, a fully view of the discipline in the PMBOK Guide (Project Management Body of Knowledge, PMBOK) that is generally recognized as good guideline, and one more in line with managing for stakeholder success and value creation. From the view point of practice, PMBOK emphasizes the importance of project delivery effectiveness as well as 'on time, in budget, to scope' efficiency.

In essence, it is a challenge for any project organization to identify of critical knowledge and the ability for utilization. Projects are variously defined but the key element is that it is a one-off task that has a finite completion. Otherwise the task is termed a program. As a result of this change in management practices there are a huge

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 120–130

Received 19/07/07
Accepted 31/08/07
Final version 20/09/07

variety of project managers in both the public and private sector who are managing projects that can range across the organizational spectrum.

In project developing stage, may with the complex and dynamic nature of problem solving methods, as PMBOK [22] defined, a project is a temporary endeavor undertaken to create a unique product, service, or result. Furthermore, progressive elaboration is a characteristic embedded in projects; it means developing in steps, and continuing by increments. Therefore, knowledge reusing and concept sharing are coming to the surface. Given the strike progress of project development technology associated with the continued rapid growth in knowledge management, it is imperative that we investigate the property of problems solving knowledge of the requirements of each stage in PM operations and try to design its ontology, that is, task ontology, accessible by different project teams in order to reduce the inconsistent on project development.

2 Related Work

2.1 Knowledge Management

In the past decade, perhaps, the most dramatic evolution, a new agenda, in business is the dawn of the new economy [9]. A hallmark of the new economy is the ability of organization to increasingly recognize that in the post-industrial era, an organization success is determined mainly by economic value from their collection of intellectual assets as well as their assets of information, production distribution, and affiliation.

In order to achieve competitive sustainability, many organizations are launching extensive knowledge management efforts and relying heavily on knowledge creation. Unfortunately, due to lack of absorptive capacity, many knowledge management projects are, in reality, information projects [24]. Musen [19] pointed out that one of the major shortcomings of the current technology for knowledge based building is lack of reusability and sharability of knowledge. This makes it difficult to build knowledge bases, since one always has to build them from scratch, "what he/she believe" and ignore the actual and potential resources embedded within, available through, and derived from the "justified true believe". Clearly, facilitating knowledge usable and useful thus should contribute to making it easier to build knowledge bases and to fit to the use-context. In order to achieve this, Mizoguchi et al. [18] indicated that expertise could be decomposed into a task-dependent but domain-independent portion and a task-independent but domain-dependent portion. The former is called task knowledge, formalized the knowledge for problem solving domain-independently.

2.2 Task Ontology

At the end of the 20th century and the beginning of the 21th, ontologies have emerged as an important and increasing interest research area in a variety of academic setting.

This phenomena stem from both their conceptual use of organizing information and their practical use in communicating about system characteristics [8].

In general, an ontology can be viewed as an information model that explicitly describes the various entities and abstractions that exist in a universe of discourse, along with their properties [11]. Moreover, an ontology is a partial specification of a conceptual vocabulary to be used for formulating knowledge-level theories about a domain of discourse. From system standpoint, ontologies provide an overarching framework and vocabulary to describe system components and relationships for communicating among architecture and domain areas [7]. Therefore, the more the essence of things is captured, the more possible it is for the ontology to be shared [10].

There is a number of categorization of ontologies currently in place. Van Heijst et al. [25] classified ontologies according to the amount and type of structure of the conceptualization and the subject of the conceptualization. While, Guarino [12] distinguished the type of ontologies by their level of dependence on a particular task or point of view. Later on, Lassila and McGuinness [14] grouped ontolgies from the perspective of the information the ontology need to express and the richness of its internal structure.

Ontologies and problem solving methods (PSMs) have been created to share and reuse knowledge and reasoning behavior across domains and tasks [10]. Benjamins and Gomez-Perez [1] defined PSMs as a way of achieving the goal of a task. It has inputs and outputs and many decompose a task into subtask, and tasks into methods. In addition, a PSM specifies the data flow between its subtask.

Given the definition from Guarino [12], task ontology is an ontology formally specifying the terminology associated with a problem type, a high-level generic task which characteristics generic classes of knowledge-based application. Chandrasekaren [5] also defined task ontology as " a base of generic vocabulary that organizes the task knowledge for a generic task". From the problem solving viewpoint, Newell [21] illustrated that task ontology can be used to model the problem solving behavior of a task either at the knowledge level or symbol level.

Thus, the advantage of task ontology is that it specifies not only skeleton of the problem solving process but also context where domain concepts are used . In 1995, Mizoguchi, Tijerino, and Ikeda [17] developed a task analysis interview system, MULTIS, which interacts with domain experts to identify the detailed task structure based on two level mediating representations. Ikeda, Seta, Kakusho, and Mizoguchi [13] indicated that task ontology can be interpreted in two ways: (1) Task-subtask decomposition together with task categorization, and (2)An ontology for specifying problem solving processes. They developed CLEPE, based on task ontology, to make problem solving knowledge explicit and exemplify its availability. Rajpathak and Motta [23] formalized task ontology by using the OCML which provides both support for producing sophisticated specifications, as well as mechanisms for operational definitions to provide a concrete reusable resource to support knowledge acquisition.

With the volumes of information continue to increase rapidly, the task of turning and integrating this resource dispersed across web resources into a coherent corpus of interrelated information has become a major problem. The emergence of the Semantic Web has shown great promise for the next generation of more capable information technology solutions and marked another stage in the evolution of ontologies and PSMs

[10]. Berners-Lee [2], inventor of the web and coiner of the term "Semantic Web", depicted that the Semantic Web is envisioned as an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in corporation, effective inter-weaving human understanding of symbol with machine processability. The way to fulfillment of the corporation can be paved by using shared knowledge-components, and so ontologies and PSMs have emerged as a core technology for developing the Semantic Web.Semantic Web with ontologies and PSMs can solve some problem much more simple than before and make it possible to provide certain capabilities they have otherwise been very difficult to support [6].

3 Research Methodology

In this section we present the TTIPP (Task analysis, Task ontology, IDEF0, Petri net, PNML) framework, shown as Fig.1, to organize and model the task knowledge acquire during the knowledge acquisition activity, using external resources and implement XML-based languages in which the task ontology will be formalized and implemented. TTIPP framework aimed at not only reducing the brittle nature of traditional knowledge-based system, but also enhancing the knowledge reusability and sharability over different applications. Furthermore, based on Rajathak et al. [23] suggestions, three important issues, including appropriate level of generality, domain independent knowledge representation, domain expert perspicuity, were considered while developing the task ontology. Also, according to the analogy of natural language, TTIPP was composed three layers and five phases. The top layer is called "lexical level model" mainly deals with the syntactic aspect of the problem solving description in terms of the task analysis phase and task ontology phase were presented.The middle layer is called "conceptual level model" captures conceptual level meaning of the description, IDEF0 model phase and Petri net model were shown in this layer. The bottom layer is called "symbol level model", with PNML phase, corresponds to runnable program and specifies the computational semantics of the problem solving. Armed with the above mentioned points, this study can provide a core epistemology for the knowledge engineer while developing the task ontology for the generic task. Now, we present the research framework model for illustrating the important phases in the development of the task ontology.

Phase-I: The Task Analysis. During the first phase of development the nature task needs to be analyzed thoroughly at a fine-grained level with diverse information needs. The structure, semi-structure, or even unstructured knowledge could be acquired and elicited from the various sources such as, the available literature on the task, the test cases specific to the problem area, the actual interview of the domain experts, the previous experience in the field etc. Ikeda et al. [13] pointed out that task analysis is made according to two major steps: (1) rough identification, and (2) detailed analysis. Based on the various sources of knowledge, rough identification of task structure is a classification problem and detailed task analysis however is to interact with domain experts and articulate how they perform their task. Once the various knowledge sources, in terms of a variety of forms (document, fact, record,...) are analyzed in detail

then the important concepts from all the different classes of application can be a heightened awareness in such a way that this knowledge provides enough theoretical foundation for expressing the nature of the problem. According to this view, concentrated on the most important concepts around which the task ontology needs to be built is the initial focus of the task analysis.

Phase-II: Conceptualization of the Task Ontology. Detailed level of the concept is indispensable for task knowledge description. Thus, this stage is generic in the sense that it gives the fundamental understanding about the relations among different concepts. Also, in according with the elicited concepts given in the previous phase, this stage provides the knowledge or ontological engineer an idea about the important axioms that needs to be developed in order to decide over the competence of the task ontology. From the standpoint of granularity and generality, Ikeda et al.[13] suggested that lexical level task ontology should consist of four concepts: (1) Generic nouns representing objects reflecting their roles appearing in the problem solving process, (2) Generic verb representing unit activities appearing in the problem solving process, (3) Generic adjective modifying the objects, and (4) Other words specific to the task.

Phase-III: IDEF0 Model. During this phase, task ontology in the research framework can be operationalized by using the formal modeling language tool. It transforms the concepts described at the natural language level into the formal knowledge modeling level in terms of structured graphical forms. Multi-level model with different classes and relations can be created in order to formalize the complex problem into being simple and more detailed to understand at each individual level based on its input parameters. Thus, the input parameters, altered by the activity or function, identified at each level can be modeled in such a way that the expected output to the problem can be achieved. IDEF0 is an activity-oriented and has been widely used modeling approach [9]. Its diagram based on a simple syntax, as showed in Fig.3, contains of an ordered set of boxes representing activities performed by the task. The boxes call ICOM' Input – Control – Output - Mechanism – are hierarchically decomposed continues until there is sufficient in detail on the basic activities to serve the tasks [24].

Phase-IV: Petri net Model Broadly speaking, the IDEF0 has a number of disadvantages in terms of its time-based function, including cumbersomeness, ambiguity in activity specification, and perhaps most significantly, its static nature [6]. Petri nets (PN) has emerged over the last ten years as a powerful tool especially suitable for systems that exhibit concurrent, conflict, and synchronization [15]. A Petri net necessarily consists of three entries: 1) the place, drawn as a circle, 2) the transition, drawn as a bar, and 3) the arcs, connecting places and transitions, as shown in Fig.4(a) [7]. Generally, the PN is defined as follows [4]:

PN = (P, T, A, W, M0) *where,*

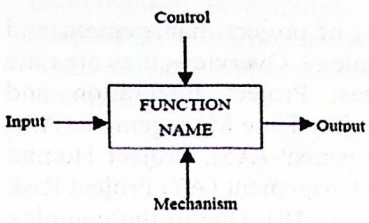
P = {P1, P2, …, Pm} *is the finite set of places;*

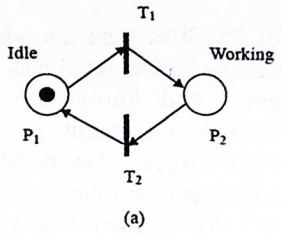
T = {T1, T2,…, Tn} *is the finite set of transitions with* $P \cup T \neq \emptyset$ *and* $P \cap T = \emptyset$*;*

A= {P x T} ∪ {T x P} *is the set of arcs between The places and transitions;*

W:A → {1, 2, 3,...} *is the weight function on the arcs;*

M0 :P → {0, 1, 2, 3, ...} *is the initial marking.*





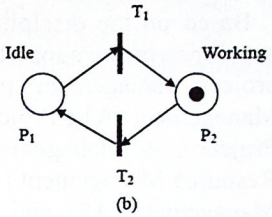


Fig.3 Component of the IDEF0 **Fig.4** Component of Petri net Model

Cassandras and Lafortune [4] further pointed out that a transition is enabled if each input place P of T contains at least the number of tokens equal to weight of the directed arc connecting P to T. Where an enabled transition T1 fires as shown in Fig.4(b), it removes the token from its input place and deposits it on its output place. Known as condition/event nets or place/transition nets, PN models are suitable to represent the structure of systems in an extensive use of hierarchical level that exhibit concurrency, conflict, and synchronization [15].

Phase-V: Petri Net Markup Language. The Petri Net Markup Language (PNML) is an XML-based interchange format for Petri nets. It is designed to be a Petri net interchange format that is independent of specific tools and platforms. Moreover, the interchange format needs to support different dialects of Petri nets and must be extensible. Thus, PNML should necessarily include the following essential characteristics [3], (1)Flexibility means that PNML should be able to represent any kind of Petri net with its specific extensions and features.(2)Ambiguity is removed from the format by ensuring that the original Petri net and its particular type can be uniquely determined from its PNML representation.(3)Compatibility means that as much information as possible can be exchanged between different types of Petri nets.

Even with a mature development on Petri net technology, it is difficult to know what is possible in the future. Certainly, PNML should shed light on the definition of Petri net types to support different versions of Petri nets and, in particular, future versions of Petri nets. Due to the above mentioned, PNML is adopted as a starting point for a standard interchange format for Petri nets.

4 Modeling PMBOK Task Ontology

Project development and management is a complex problem especially it requires collaboration and participation from strategical level to operational level staff in organization. Hence, the primary purpose of the PMBOK guideline is to identify the subset of the project management and development body of knowledge. It is a general overview which means the knowledge and practices described are applicable to most

projects most of the time, meanwhile, the guideline can lead the correct application of these skills, tools and techniques can enhance the chances of success over a wide range of different projects.

Based on the discipline of PMBOK task knowledge of project management and development, we adopt these as the lexical level task ontology. Overview, they are nine project management processes and knowledge areas: Project Integration and Management (A1), Project Scope Management (A2), Project Time Management (A3), Project Cost Management (A4), Project Quality Management (A5), Project Human Resource Management (A6), Project Communications Management (A7) Project Risk Management (A8), and Project Procurement Management (A9), Due to the complex and dynamic nature of problem solving methods for project management, risk management (A8) of debris flow was chosen to serve as an example for presenting the TTIPP framework.

At the modeling stage, the IDEF0 model is built for describing the function of the risk management. From a functional point of view, the present risk management has six activities as shown in Fig.6. The six activities are "Risk Management Planning" (A81),"Risk Identification" (A82),"Qualitative Risk Analysis" (A83),"Quantitative Risk Analysis" (A84),"Risk Response Planning"(A85) and "Risk Monitoring and Control" (A86) as shown in Fig.6. The different mechanisms support the different sub-activities as shown in Fig.6. After obtaining the functional IDEF0 model, we can then develop the Petri net model for behavior analysis at next stage.

According to the six activities in the IDEF0 model built at previous stage, the PN model is constructed. The PN model consists of 29 places and 12 transitions, of which the corresponding notations for "Risk Management"(A8) are described in Table 1 and Table 2.

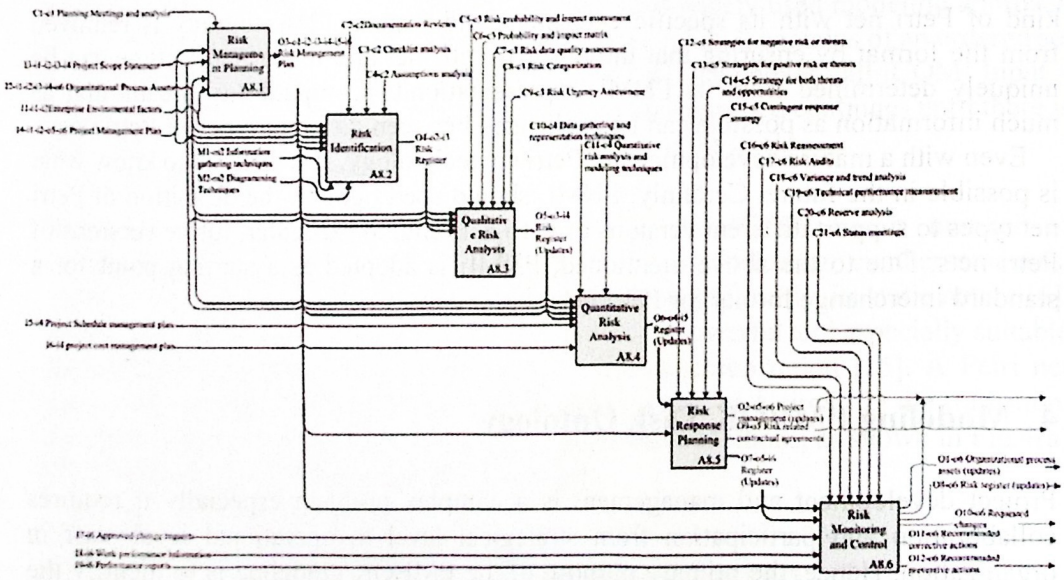


Fig.6. The IDFE0 model of the "Risk Management " (A8)

After constructing the PN model, we performed the model validation of the dynamic behavior using reachability tree. It revealed that the PN consisted of the liveness, boundedness, reversibility, and reachability. At symbol level model, PNML was adopted as a starting point for a standard interchange format for Petri nets. The XML-based interchange format for the above-mentioned Petri net model is generated automatically

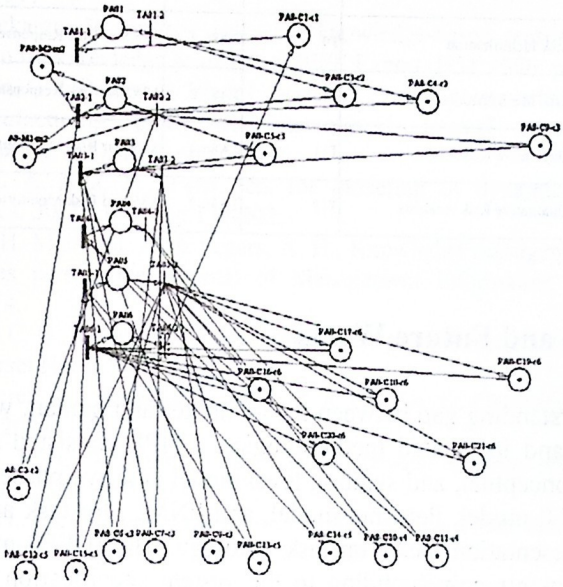


Fig.7. The PN model of the "Risk Management " (A8)

Table 1. The Notation of place for "Risk Management" (A8)

Place	Name	IDEF0	Place	Name	IDEF0
P1	PA81	A81 Risk Management Planning	P16	PA8-C10-c4	C10-c4 Data gathering and representation techniques
P2	PA82	A82 Risk Identification	P17	PA8-C11-c4	C11-c4 Quantitative risk analysis and modeling techniques
P3	PA83	A83 Qualitative Risk Analysis	P18	PA8-C12-c5	C12-c5 Strategies for negative risk or threats
P4	PA84	A84 Quantitative Risk Analysis	P19	PA8-C13-c5	C13-c5 Strategies for postive risk or threats
P5	PA85	A85 Risk Response Planning	P20	PA8-C14-c5	C14-c5 Strategy for both threats and opportunities
P6	PA86	A86 Risk Monitoring and Control	P21	PA8-C15-c5	C15-c5 Contingent response strategy
P7	PA8-C1-c1	C1-c1 Planning Meetings and analysis	P22	PA8-C15-c5	C15-c5 Contingent response strategy
P8	PA8-C2-c2	C2-c2 Documentation Reviews	P23	PA8-C17-c6	C17-c6 Risk Audits
P9	PA8-C3-c2	C3-c2 Checklist analysis	P24	PA8-C18-c6	C18-c6 Variance and trend analysis
P10	PA8-C4-c2	C4-c2 Assumptions analysis	P25	PA8-C19-c6	C19-c6 Technical performance measurement
P11	PA8-C5-c3	C5-c3 Risk probability and impact assessment	P26	PA8-C20-c6	C20-c6 Reserve analysis
P12	PA8-C6-c3	C6-c3 Probability and impact matrix	P27	PA8-C21-c6	C21-c6 Status meetings
P13	PA8-C7-c3	C7-c3 Risk data quality assessment	P28	PA8-M1-m2	M1-m2 Information gathering techniques
P14	PA8-C8-c3	C8-c3 Risk Categorization	P29	PA8-M2-m2	M2-m2 Diagramming Techniques
P15	PA8-C9-c3	C9-c3 Risk Urgency Assessment			

Table 2. The Notation of transition for "Risk Management" (A8)

Transition	Name	IDEF0	Transition	Name	IDEF0
T1	TA81-1	A81 Star Risk Management Planning	T7	TA84-1	A84 Star Quantitative Risk Analysis
T2	TA81-2	A81 End Risk Management Planning	T8	TA84-2	A84 End Quantitative Risk Analysis
T3	TA82-1	A82 Star Risk Identification	T9	TA85-1	A85 Star Risk Response Planning
T4	TA82-2	A82 End Risk Identification	T10	TA85-2	A85 End Risk Response Planning
T5	TA83-1	A83 Star Qualitative Risk Analysis	T11	TA86-1	A86 Star Risk Monitoring and Control
T6	TA83-2	A83 End Qualitative Risk Analysis	T12	TA86-2	A86 End Risk Monitoring and Control

5 Conclusions and Future Works

To bridge the understanding gap between the computer and human, we presented the TTIPP framework and its related methodologies. TTIPP consisted of three layers, including lexical, conceptual, and symbol, level model and five phases: task analysis, task ontology, IDEF0 model, Petri net model, and PNML, for task analysis problem and mediating representation based on task ontology. The IDEF0 model is used to capture the requirements corresponding to the system specification at the stage of functional analysis. Subsequently, at the stage behavior analysis, the Petri net model is constructed according to the IDEF0 model. Finally, at the implementation stage, the obtained model can be realized by using Petri net marked coding languages.

Ideally, task ontology, produced by TTIPP, does not subscribe to any specific problem solving approach it provides a sound ontological foundation for the different problem solving approaches and can be used to support a great variety of task modeling independently of the target shell or computational method. Therefore, it not only enables better access to information and promote shared understanding for humans, but also facilitates comprehension of information and more extensive processing for computers. Project development is a complex problem, because projects are undertaken at all levels of the organization and they can involve a single person or many thousands, involve one or many organizational units, such as joint ventures and partnerships. Thus, in future, we are planning to use this task ontology as a major building block for developing the generic problem-solvers for understanding the space of the PMBOK as well as advancement of the field in general.

References:

1. Benjamins, V. R. & Gomez-Perez, A. (n.d.). Knowledge-System technology: ontologies and Problem-Solving Methods, Retrieved December 13, 2004, from http://hcs.science.uva.nl/usr/richard/pdf/kais.pdf

2. Berner-Lee, T., Weaving the web: The original design and ultimate destiny of the world wide web by its inventor, HarperCollins Publishers, NY, 1999.
3. Billington, J. Christensen, S. van Hee, K. Kinder, E. Kummer, O. & Petrucci, L., The petrinet markup language: Concepts, technology, and tools, URL htto://www.informtik.hu-berlin.de/top/PNX/, 2004.
4. Cassandras, C. G. & Lafortune, S., Introduction to discrete event systems, Boston, MA: Kluwer, 1999.
5. Chandrasekaran, B, Generic tasks for knowledge-based reasoning the right level of abstraction for knowledge acquisition. IEEE Expert, 1(3), 1986, pp.23-30.
6. Colguhoun, G. J. Baines, R. W. & Crossley, R., A composite behavior modeling approach for manufacturing enterprise, International Journal of Computer Integrated Manufacturing, 9(6), 1996 pp.463-475.
7. David, R, & Alla, H., Petri nets for modeling of dynamics systems—A survey, Automatica, 30(2), 1994, pp.175-202.
8. Dold, A. H. Malhotra, A. & Segars, A. H., Knowledge management: An organizational capabilities perspective. Journal of Management Information System, 18(1), 2001, pp.185-214.
9. Feldmann C. G., The practical guide to business process reengineering using IDEF0. New York: Dorset House Publishing, 1998.
10. Gomez-Perez, A., Fernandez-Lopez, M. & Corcho, O, Ontology engineering. London: Springer, 2004.
11. Gruber, T. R. A., A translation approach to portable ontology specification, Knowledge Acquisition, 5(2), 1993, pp.199-221.
12. Guarino, N., Formal ontology and information system, 1998, Retried December 6, 2004, from http://www.loa-cnr.it/Papers/FOIS98.pdf
13. Ikeda, M., Seta, K., Kakusho, O. & Mizoguchi R., Task ontology: ontology for building conceptual problem solving models. Proceedings of ECAI98 Workshop on Applications of ontologies and problem-solving model, 1998, pp.126-133.
14. Lassila, O. & McGuinness, D , The role of frame-based representation on the semantic web (KSL-01-02). Knowledge System Laboratory, Stanford University, Stanford, CA, 2001.
15. Lee, J. S. & Hsu, P. L., An IDEF0/Petri net approach to the system integration in semiconductor manufacturing systems. IEEE International Conference Systems, Man and Cybernetics, Vol.5, 2003, pp.4910-4915
16. Mizoguchi, R., Tijerino, Y. & Ikeda, M., Task ontology and its use in a task analysis interview system – Tow-level mediating representation in MULTIS, Proceedings of the 2nd Japanese JKAW92, 1992, pp.185-198.
17. Mizoguchi, R., Tijerino, Y. & Ikeda, M., Task analysis interview based on task ontology. Expert Systems With Applications, Vol.9, No.1, 1995, pp.15-25.
18. Mizoguchi, R., Vanwelkenhuysen, J. & Ikeda, M., Task ontology of reuse of problem solving knowledge. Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing, 1995, pp.46-59.
19. Musen, M., Dimension of knowledge sharing and reuse. Knowledge Systems Laboratory (report KSL-91-65). Stanford, CA: Stanford University, 1991.
20. National Institute of Standards and Technology , Integration definition for function modeling (IDEF0). FIPS 183, 1993, Retrieved November 30, 2004, from http://www.itl.nist.gov/fipspubs/idef02.doc
21. Newell, A., The knowledge-level, Artifical Intelligence, 18, 1982, pp.87-127.
22. Project Management Institute(PMI), 2004, A Guide to the Project Management Body of Knowledge: PMBOOK Guide, Third Edition.,USA.

23. Rajpathak, D. G., The task ontology component of the scheduling library: Pilot study. Knowledge Media Institute, The Open University, 2001.
24. Santarek, K. & Buseif, I. M., FMS design using non-conventional model techniques and standard software tools. 5th natl. Conf. on Robotics, Institute of Technical Cybernetics, Wroclaw technical University, Poland, 1996.
25. Van Heijst, G., Schreiber, A. T. & Wielinga, B. J., Using explicit ontologies in KBS development. International Journal of Human-Computer Studies, Vol. 46, Issue: 2-3, 1997, pp. 183-292.

Analysis and Classification of Car Engine Sounds to Diagnose Failures

Luis Andrés Méndez-Ortega[1] and Orion Fausto Reyes-Galaviz[2]

Universidad Autónoma de Tlaxcala
Facultad de Ciencias Básicas, Ingeniería y Tecnología
Apizaco. Tlaxcala. México
[1]siul222@hotmail.com,
[2]orionfrg@ingenieria.uatx.mx
http://ingenieria.uatx.mx/

Abstract. The detection of a possible car engine failure can be achieved through the identification on the vibration changes that are produced while the motor is operating. A person with enough mechanical experience is capable of recognizing a flaw in an engine, just by listening to the noise and sensing the vibrations produced. A car owner can notice a flaw in the motor, when a shift on the movement of the engine is felt, or if there's an abnormal noise being produced by the same. Nevertheless, it's hard to guess what kind of malfunctions produce certain motors noises and vibrations, making it hard to work directly on the engine's flaw. In this work, an interface is designed to classify, and directly detect, four common car engine failures, just by using the engine's recorded sound. To accomplish this goal, we use Linear Prediction Cepstral Coefficients (LPCC) and Mel Frequency Cepstral Coefficients (MFCC) as the main acoustic characteristics extractors, and a Time Delay Neural Network for the pattern recognition. Some results are shown of up to 95.55% when predicting common car failures.

Key words: Engine noise, LPCC, MFCC, Time Delay Neural Networks, Expert System

1 Introduction

A failure on any engine can produce mild or violent vibrations that can damage the motor, if left unattended. These vibrations can cause fissures, waste and/or overheating of important parts and reduce the machine's performance. Besides, the vibrations are a good indicator of the mechanical performance and they are also very sensitive to the malfunction evolution. In general, these types of flaws are preceded by shifts on the sound conditions, vibration, power loss, etc. These indicators are a sign of some kind of future failure on the engine's performance. Although, not all the sources of sounds or vibrations are inevitable, due to the fact that some are natural to the machine's operation by it self, enhancing with this the importance of identifying which sound/vibration corresponds to a possible malfunction. Most of the sounds that an engine produces are generated by

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 131–140

Received 13/06/07
Accepted 31/08/07
Final version 28/09/07

rotating mechanisms inside the motor; also, the machines that have an internal combustion system, produce more noise due to the explosions created inside.

In general, a trained ear can quickly detect the existing differences between the noises that are called normal, and a series of abnormal noises produced by vibrating gears, pounding on the engine, hisses, etc. If this kind of relevant acoustic information exists, hidden inside the engine's noise, the extraction, recognition, and classification of the acoustic characteristics could be possible, using automatic mechanisms, in order to obtain reliable diagnostics.

The engine noise can be an important clue to the prediction of possible failures, and can be helpful to preserve a good motor performance. The analysis presented in this work, joint with the monitoring systems that actually exist, can produce more tools that could help the motor specialists obtain robust diagnostics when repairing car engines. Taking this into account, an interface is designed to capture and analyze the acoustic sound produced by these kind of machinery, some acoustic characteristics extraction techniques are used, and pattern recognition, in order to classify the different types of noises that are produced by engines with good and bad performance. On the next section we will present a review of prior comparative studies on the field. Section 3 details the fundamental basis in the noise recognition process and describes our proposed system. Section 4 deals with acoustic processing and the feature extraction method which uses the Linear Prediction Cepstral Coefficients (LPCCs) and Mel Frequency Cepstral Coefficients (MFCCs) methods. A fundamental theory on pattern classification, and Time Delay Feed-Forward Neural Networks, is given in Section 5. The complete system implementation, including the user interface, can be found on Section 6. Our experimental results, that go up to 95.58%, are shown on section 7, and the concluding thoughts are presented on Section 8.

2 State of the Art

Many systems have been proposed on the acoustic pattern recognition field in the past few years. Depending on the problem that has to be solved, a common point of some approaches is focused on the detailed spectral analysis of the acoustic signal. In [1] Mario E. Munich uses Mel Frequency Cepstral Coefficients (MFCC) for the acoustic analysis, Gaussian Mixture Models (GMM), and Hidden Markov Models, combined with Bayesian Subspace Methods, applied to the automatic recognition of acoustic characteristics from vehicles; this is used on military operations, for surveillance purposes, achieving a precision of 83%. Huadong Wu, Et Al [2], also worked on the vehicle recognition field. They proposed that each vehicle model has the same kinds of noises, vibrations, hops, and tire friction. They used a method called eigenfaces, used most commonly on the face recognition field, to characterize the noise patterns and use them to recognize the vehicle; this method is also known as the Korhunen-Loeve expansion or as the Principal Component Analysis (PCA), te results on this work seem promising. In [3], Edgar A. Estipiñán, Et Al use an analysis of mechanical

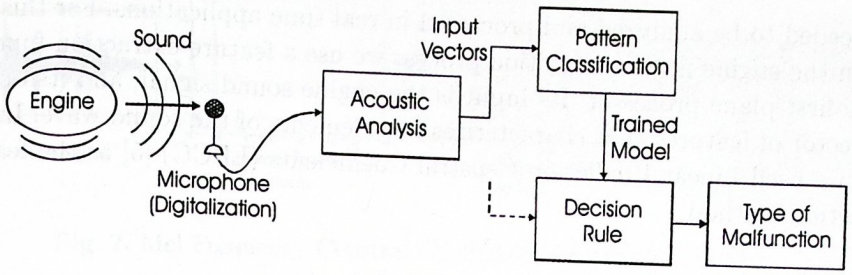


Fig. 1. Engine Noise Recognition Process

vibrations, as a part of their predictive maintenance, to establish the mechanical health of their machines, preventing future flaws with this technique. They propose techniques based on the Fast Fourier Transform to analyze the noise produced by the vibration of slow rotatory machinery. Pedro N. Saavedra [3], [4], presents some techniques applied to the vibration analysis, he particularly analyzed the use of fissure detection on machinery axis and rafters. For the theoretical study, the Finite Element was used, and the fissure was modeled using lineal fractomecanical theory.

It is difficult to compare these systems; because the data bases, experimental conditions, number of samples per second, frame size, and type of recognitions, are different. Although, the analysis of different techniques, suggests interesting possibilities of combining them in order to obtain better recognition results.

3 Car Engine Noise Recognition Process

The car engine noise recognition process is basically a pattern recognition problem, and it is similar to speech recognition. The goal is to take the motor's sound wave as an input, and at the end recognize the engine's malfunction. Generally, the engine noise recognition process is done in two steps; the first step is the acoustic processing, or features extraction, while the second is known as pattern processing or classification. The proposed system can be seen in Fig. 1. For this case, in the acoustic analysis, the engine's signal is processed to extract relevant features in function of time. The feature set obtained from each noise sample is represented by a vector, and each vector is taken as a pattern. As for the pattern recognition methods, four main approaches have been traditionally used: pattern comparison, statistical models, knowledge based systems, and connectionists models. We focus on the use of the last one.

4 Acoustic Processing

The acoustic analysis implies the application and selection of filter techniques, feature extraction, signal segmentation, and normalization. With the application of these techniques the signal is described in terms of its fundamental components. An engine signal is too complex and codifies more information than the

one needed to be analyzed and processed in real time applications. For this reason, in the engine noise recognition process we use a feature extraction function as the first plane processor. Its input is the engine sound signal, and its output is a vector of features that characterizes key elements of the sound wave. In this work we used Linear Prediction Cepstral Coefficients (LPCC) [5] as the feature extraction method.

4.1 Linear Prediction Cepstral Coefficients

The Lineal Prediction (LP) method is historically one of the most important methods used for the voice analysis [5]. Its fundamental basis is to establish a filter model for the sound source. With enough number of parameters, the LP model can establish a suitable approximation to the spectral structure of any kind of sound. This is why this technique is used to analyze our acoustic samples. The LP method gets this name because it pretends to extrapolate the value of the sound's next sample $x(n)$ as the weighted sum of the previous samples $x(n-1), x(n-2), ..., x(n-k)$:

$$x(n) = \sum_{i=1}^{K} a_i x(n-1) \tag{1}$$

To do so, a coefficient computation must be made, by minimizing an error function E, specifically the least square, over a window of size N.

$$\begin{aligned} E &= \sum_{n=0}^{N-1} e^2(n) \\ &= \sum_{n=0}^{N-1} \left(x(n) - \sum_{i=1}^{K} a_i x(n-1) \right)^2 \quad 0 \leq n \leq N-1 \end{aligned} \tag{2}$$

Departing from the LP analysis, it's possible to obtain the associated cepstral coefficients expression (LPCC):

$$c(0) = \log(1) = 0 \tag{3}$$

$$c(i) = -a(i) - \sum_{j=1}^{i-1} \left(1 - \frac{j}{i}\right) a(j) c(i-j) \quad 1 \leq i \leq N_c \tag{4}$$

A common transformation, over this kind of coefficients, is known as the delta cepstral coefficients or delta cepstrum coefficients. We can obtain these by applying the next expression:

$$\Delta c_j(i) = \frac{1}{2T+1} \sum_{k=-T}^{T} k \cdot c_{j+k}(i) \tag{5}$$

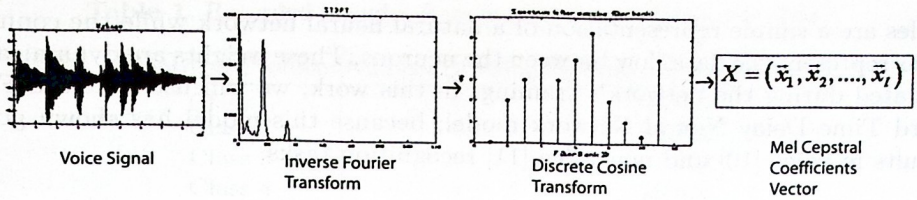


Fig. 2. Mel Frequency Cepstral Coefficients extraction method.

4.2 Mel Frequency Cepstral Coefficients

The first step of the sound processing, once we have the noise samples, is to obtain (from these samples) the spectral characteristics. This step is necessary because the important information of the sample is codified in the frequency domain, and the engine noise samples are recorded by means of electronic devices in the time domain. When the time domain is converted to the frequency domain we obtain the parameters which indicate the occurrence of each frequency.

There is a wide variety of ways to represent the noise samples in their parametric form. One of the most commonly used on speaker recognition tasks are MFCCs. The human ear decomposes the received sound signals in its fundamental frequencies. Located in the inner ear we find the cochlea which has a conic spiral form. This is one of the three cavities that form the physical structure of the ear [6]. This cochlea filters the frequencies in a natural way. The sound waves are introduced inside this structure bouncing on its walls and getting inside the spiral with low or high frequency, taking into account each frequency's wave length [7]. MFCCs are based on the frequency response the human ear perceives. This method behaves as a filter bank linearly distributed in low frequencies and with logarithmic spacing on the higher frequencies. This is called the Mel Frequency Scale, which is linear below 1000 Hz, and logarithmic above 1000 Hz (Fig. 2) [8].

5 Engine Noise Pattern Classification

After extracting the acoustic features of each noise sample, the feature vectors are obtained; each one of these vectors represents a pattern. These vectors are later used for the classification process. For the present work, we focused on connectionist models, also known as artificial neural networks (ANN), to classify these vectors (patterns).

5.1 Neural Networks

Artificial neural networks (ANN) are widely used on pattern classification tasks, showing good performance and high accuracy results. In general, an artificial neural network is represented by a set of nodes and connections (weights). The

nodes are a simple representation of a natural neural network while the connections represent the data flow between the neurons. These weights are dynamically updated during the network's training. In this work, we will use the Feed Forward Time Delay Neural Network model; because this model has shown good results in voice [10] and non-voice [11] recognition tasks.

5.2 Feed-Forward Time Delay Neural Network

Input Delay refers to a delay in time, in other words, if we delay the input signal by one time unit and let the neural network receive both the original and the delayed signals, we have a simple time delay neural network. This neural network was developed to classify phonemes in 1987 by Weibel and Hanazawa [12].

The Feed-Forward Time Delay neural network doesn't fluctuate with changes; the features inside the sound signal can be detected no matter in which position they are in. The time delays let the neural network find a temporal relation directly in the input signal and in a more abstract representation of the hidden layer. It does this by using the same weights for each step in time [10].

5.3 Scaled conjugate gradient back-propagation

The neural network training can be done through a technique known as back-propagation. The scaled conjugate methods are based on the general optimization strategy. We use scaled conjugate gradient back-propagation (SCGBP) to train the neural networks; because this algorithm shows a lineal convergence on most of the problems. It uses a mechanism to decide how far it will go on a specific direction, and avoids the time consumption on the linear search by a learning iteration, making it a fast second order algorithm [9].

6 System Implementation

In most towns and cities in the Mexican territory, the use of public transportation is very common. A car model, that is commonly used for this purpose, is the *Combi Van* from Volkswagen, models '82, '83, '86, '88, and '90. These cars work almost every day from 6:00 to 21:00, and there are some hours that they carry up to 18 persons and travel several kilometers with this weight. Some car shops, in cities where these vehicles are used as public transportation, specialize on maintaining and repairing only these kind of engines, giving the engine specialists, the ability to quickly diagnose most of the common motor flaws, and directly work on the problem of these specific engines. For these reasons we choose this kind of engines. The conditions to select and record an engine were:

- Old models or engines previously repaired.
- Similar engine characteristics between engines.
- Several similar car models, or with the same engine.

Table 1. Recorded samples from accelerating engines, per class

Identified Problem	Number of Samples	Time (in seconds)
Class 1	7	18.036
Class 2	6	18.502
Class 3	6	17.664
Class 4	7	18.006
Total	26	72.208

We recorded the engine's sound in two phases; when the car first arrived to the shop, and after the car was diagnosed and repaired. This method helped us label the recordings and compare the noise produced by the engine before and after the service. The two phases consisted where; accelerating the engine, and in *Ralenti* state (no acceleration). Also, some noises were recorded from tuned engines in both phases.

6.1 Noise Samples

To record the engine noise, we visited car shops recommended by the public transportation drivers. The recorded samples were obtained at five different car shops, three of them specialized only on Combi Vans. Each recorded sample was obtained with the help of a ZicPlay digital audio recorder, and the output files were saved in WAV format with the following settings; 8 Khz, 16 bits, and monaural channel. Each of these files has a time period of 3 to 5 seconds,and a total of 26 different engines were recorded. These samples were classified in four categories, tuned engines (Class 1), crank related problems (Class 2), piston related problems (Class 3), and valves related problems (Class 4), see Table 1. On Table 2 we can see the number of samples obtained from engines on Ralenti state.

It has to be taken into account that these samples were difficult to obtain; in some cases some samples had to be rejected, due to the poor sound quality, noise contamination, and other issues that made the sample unusable for the system's goals. 3 different flaws were used, plus one from tuned/fixed motors, because according to the experts, these were the most common flaws. At first, four common flaws were considered, but two of them directly related to a single common flaw, leaving us with only four different classes.

Table 2. Recorded samples from engines in Ralenti state, per class

Identified Problem	Number of Samples	Time (in seconds)
Class 1	7	16.396
Class 2	6	15.430
Class 3	6	15.744
Class 4	7	16.590
Total	26	64.208

The sound recordings, obtained from engines in both, accelerating and Ralenti states, where segmented in 1 second samples and labeled from classes 1 thru 4; for each segmented sample we extracted different configurations of LPCC and MFCC (as shown on Tables 3 and 4). The LPCC and MFCC configurations were obtained heuristically, and were the ones that gave the best overall results.We used 80% of the segmented samples to train the NN and 20% to test it. The network architecture consists on; an input layer with the number of nodes corresponding to the input vectors size, a hidden layer with 60% less nodes than the input layer, and an output layer corresponding to the number of classes to predict, in this case 4. The NN was implemented with the help of the Neural Network Tool-box, in Matlab 7.0, and the characteristic extraction was done with the help of the Auditory Tool-box for Matlab [13].

7 Experimental Results

On Table 3 we can see the results obtained by using the samples from engines in Ralenti state. Table 4 shows the results obtained from samples recorded from accelerating engines. Each result is the best overall result from 10 experiments. On both tables, the first column shows the number of characteristics extracted (c) per each time frame (ms).

Table 3. Results obtained with samples from engines in Ralenti state.

Configuration	Precision	Configuration	Precision
LPCC 38c 50ms	80.77%	MFCC 38c 50ms	80.00%
LPCC 36c 100ms	95.41%	MFCC 36c 100ms	85.60%
LPCC 26c 100ms	95.22%	MFCC 26c 100ms	82.97%
LPCC 20c 100ms	89.74%	MFCC 20c 100ms	75.79%

The results on both tables show that the best precision was given when using 36 LPCCs for each 100ms time frame. Taking this into account, we randomly selected 12 WAV files from the training samples, previously separated, and fed them to the trained NN, calculating with this the precision percentage given by each output node. The results obtained are shown in Table 5.

Table 4. Results obtained with samples from accelerating engines

Configuration	Precision	Configuration	Precision
LPCC 36c 100ms	95.55%	MFCC 36c 100ms	84.85%
LPCC 26c 100ms	93.01%	MFCC 26c 100ms	88.35%
LPCC 20c 100ms	86.48%	MFCC 20c 100ms	83.81%

Table 5. Comparison between the predicted class and the real class

Test	Class 1	Class 2	Class 3	Class 4	Real Class
1	94.59	0.0	5.40	0.0	Yes
2	100.0	0.0	0.0	0.0	Yes
3	89.18	0.0	0.0	10.81	Yes
4	0.0	75.67	21.62	2.70	Yes
5	5.40	94.59	0.0	0.0	Yes
6	2.70	67.56	8.10	21.62	Yes
7	0.0	0.0	97.29	2.70	Yes
8	0.0	5.40	86.48	8.10	Yes
9	0.0	0.0	2.70	97.29	No (C3)
10	0.0	0.0	0.0	100.0	Yes
11	0.0	0.0	8.10	91.89	Yes
12	0.0	0.0	8.10	91.89	Yes

8 Conclusions and Future Work

The results show that an engine in bad conditions does in fact emit different sounds than a tuned motor, also that they can be recognized by the TDNN, depending on the type of flaw that's affecting the motor efficiency. This information can help a motor specialist give fast diagnosis, prices estimates for the reparation, and directly work on the motor's problem. We also concluded that is viable to work on this problem and build a cheap system that can be used by motor manufacturers and car shop owners.

We are working with other acoustic characteristic extractions such as Wavelets and LPC, to compare the shown results with these analysis, and make a trustworthy system. We are also collecting more sound samples mainly from Combi Van, Chevy motors, and from other car brands, to build a system that is able to recognize other engines and other failures. We also want to design a Matlab-independent user-interface that can be used on car shops, and predict engine failures, only by feeding directly the motor noise, or a pre-recorded sample.

References

1. Mario E. Munich. Bayesian Subspace Methods for Acoustic Signature Recognition of Vehicles. Published at the 12^{th} European Signal Processing Conference (EUSIPCO-2004), Vienna, Austria. September 6-10, 2004.

2. Huadong Wu, Mel Siegel, and Pradeep Khosla. Vehicle Sound Signature Recognition by Frequency Principle Component Analysis . The Proceedings of IMTC'1998, selected in the IEEE Transaction on Instrumentation and Measurement Vol. 48, No. 5, page 1005 - 1009, (ISSN 0018-9456), October 1999.
3. Pedro N. Saavedra and Edgar A. Estupiñan. Analisis De Vibraciones Aplicado A Las Maquinas Rotatorias De Baja Velocidad. internal.dstm.com.ar/sites/mmnew/bib/notas/Estupi1.pdf. Chile 2000.
4. Pedro Nelson Saavedra. La Medición y Análisis de las Vibraciones como Técnica de Inspección de Equipos Y Componentes, Aplicaciones, Normativas Y Certificación. Facultad de Ingeniería - Universidad de Concepción, Casilla 160 - Concepción - Chile. http://www.aaende.org.ar/sitio/biblioteca/material/ CONFCHILE.pdf. 2001.
5. Agustín Álvarez Marquina. Fundamentos del Reconocimiento Automático de la Voz, Algoritmos de Extracción de Caracerísticas. Facultad de Informática. Universidad Politécnica de Madrid (UPM). November 22, 2004.
6. A. J. Balagué, Diccionario Enciclopédico Baber, Editorial Baber, 1119 E. Broadway Street Glendale, California 91205. (1991).
7. Bernal J., Bobadilla J., Gómez P., *Reconocimiento de voz y fonética acústica*, Ed. Alfaomega Ra-Ma , Madrid, España. (2000).
8. Miramontes de León G., De la Rosa Vargas J. I. and García E., 'Application of an Annular/Sphere Search Algorithm for Speaker Recognition', Proceedings of the 15th International Conference on Electronics, Communications and Computers, 0-7695-2283-1/05 IEEE. CONIELECOMP, (2005).
9. Orozco-Garca, Jos. Extraccin y Anlisis de Caractersticas Acsticas del Llanto de Bebés para su Reconocimiento Automático Basado en Redes Neuronales. Tesis. INAOE, México. February 2004.
10. Hilera J., Martínez V. Redes Neuronales Artificiales Fundamentos, modelos y aplicaciones. Alfaomega, Madrid, España. (2000).
11. Reyes-Galaviz O. F., Reyes-Garca, C.A., Infant Cry Classification to Identify Hypoacoustics and Asphyxia with Neural Networks, in LNCS 2972: Advances in Artificial Intelligence. Springer, Berlin, 2004, pp 69-78, ISBN 3-540-21459-3, ISSN 0302-9743.
12. A. Weibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme Recognition Using Time Delay Neural Networks," IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-37: 32'339, (1989).
13. Auditory Toolbox. Malcolm Slaney, Interval Research Corporation http://rvl4.ecn.purdue.edu/ malcolm/interval/1998-010/AuditoryToolbox.zip

Study on the Multiagent Based Model System and Model Integration*

Daiping Hu[1], Ruiming Wu[1], Mengyu Li[2], Qi Wang[3]

[1] Antai College of Economics & Management, Shanghai Jiaotong University, 200052 Shanghai, China
{dphu, dwurm }@sjtu.edu.cn
[2]Shanghai Xiandai Vocational & Technical School
Huayan Road 112, 200042 Shanghai, China
dmyli@163.com
[3] Graduate School, Lanzhou University, Lanzhou, 730000 Gansu, China
qiwanglzu@126.com

Abstract. Model integration is to combine some simple models togather for decision making of complex problems. But the model integration is difficult to implement in the traditional DSS. In this paper, we propose a multiagent based model system and study the method of model integration using agents. The multiagent based model system includes a management agent, many interface agents and a lot of model agents. Every model is encapsulated by an agent, which can be invoked by users utilizing interface agents or other agents. The model integration method is implemented by using the cooperation of model agents. We also study a case to demonstrate the applicaton of model integration using the multiagent based model system we developed.

Keywords: Model system, model integration, multiagent system

1 Introduction

A model is a simplified representation or abstraction of reality. As reality is often too complex to copy exactly or much of the complexity is actually irrelevant in solving the specific problem, simplification is necessary. According to the degree of abstraction, models are classified into three groups: iconic, analog, and mathematical [1]. An iconic model is a physical replica of a system, usually on a different scale from the original. An analog model is a symbolic representation of reality. Representation of the complexity of relationships in many organizational systems can not be done by using icon or analog models. The most abstract model called mathematical model is often used to describe them mathematically.

We build mathematical models to represent for decision making of complex problems. Computers with the capability of speed calculation are applied in modeling. The decision support system (DSS) as a kind of computer based information system

* Supported by National Nature Science Foundation of China (No.70671067) and Youth Research Fund of Antai College of Economics & Management

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 141–149

Received 18/06/07
Accepted 31/08/07
Final version 01/10/07

(CBIS) has been studied and developed to support decision making for solving complex problems. A DSS uses data and models, provides easy user interface, and can incorporate the decision maker's own insights. Architecture of a DSS typically includes: data management subsystem, model management subsystem, knowledge management subsystem and user interface subsystem. The model management subsystem is the most important part, which is composed of model base, model base management system, modeling language, model directory, model execution, integration, and command processor.

Models are most important components of a DSS. Model integration provides the method of using two or more models or creates a new model by combining them. Integrated modeling environment should be provided in model subsystem. Some progresses have been made on model integration and some modeling integration platforms have been developed. [2].

Recent years, as achievements of research on the theories, methodologies, architectures, languages of multiagent system (MAS) accumulated, the applications of agents emerge out more and more. Multiagent based modeling applied in complex system is mainly the multiagent based simulation, a bottom-up modeling method. MAS based system as tools for dealing with complex problems is applied in the design paradigm of building agent based DSS, hall of workshop for meta-synthetic engineering (HWME) [3,4,5].

In this paper, we study the multiagent based model system and the methods of model integration. This paper is organized as follows: In section 2, methods of model integration are discussed. Section 3, multiagent based integrated modeling platform is presented. In section 4, we introduce an application about integration of forecasting models and a linear programming model. In section 5, conclusions and discussion are given.

2 Methods of Model Integration

Many researchers have studied on model integration and have made some achievements [2]. Based of their study, model integration can be classified in three ways.

(1) Scheme integration and process integration [6]. Schema integration is the task of merging internal structure of two models to create a new model. Process integration is about solver control.

(2) Definitional integration and procedural integration [6]. Definitional and procedural are in accord with the classical dichotomy of programming languages. Definitional integration is about mode representation and corresponds to scheme integration. Procedural integration is about model manipulation and corresponds to process integration.

(3) Deep integration and functional integration [7]. Deep integration is combining two or more given models together to produce a single new model, subjects to the important qualification that the new model must be represented in same definitional formalism as the given models. Functional integration does not yield a new model. It superimposes a computational agenda for coordinating calculations over the original

models by directing certain models' outputs to other models' inputs while specifying the order of computations.

To implement functional model integration by using agents, we analyze the linkage of models. Two basic and an extended modes of connections are summarized. Two basic modes are pipeline connection and parallel connections.

In pipeline connections of models, the outputs of one model are the inputs to another. The later cannot begin modeling until the previous model gives the outputs. See Fig.1.

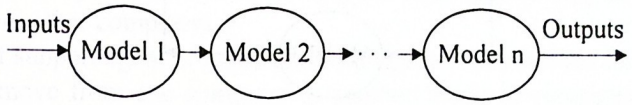


Fig. 1. Pipeline connections of models

In parallel connections of models, inputs of all the models are from the same information sources. Their inputs may be the same or may be different. Generally, if all the models can model the problem individually, their inputs may be the same. If each of the models can model a different part of the given problem, its inputs may be different. See Fig.2.

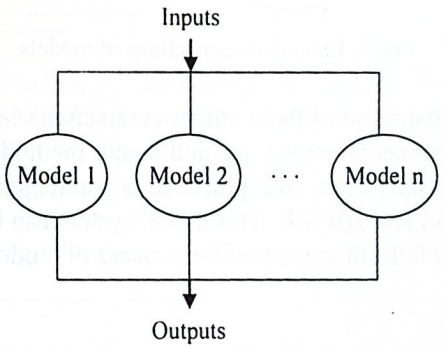


Fig. 2. Parallel connections of models

Extended connection of models may be used when there are three or more models to be integrated. It is the combination of pipeline and parallel connections of models. See Fig.3.

Agent, we mean the software agent here, as a special kind of computer program, can accomplish specific tasks on behalf of its user, independently or with little guidance. Architecture of an agent should include task processing method, knowledge, and the capabilities of task processing, learning, reasoning, decision-making, communication, cooperation etc.

A MAS is a system comprised of some agents, and all the agents can communicate with each other and for cooperation in procedure of performing tasks. According to whether the MAS if pursuing a specific system goal or not, two types of MASs can be distinguished: The specific goal and non-specific goal. A specific MAS has a specific goal, all agents in the system serve for the system goal, and each agent performs different tasks and can cooperate with others. Multiagent based DSS is a specific

goal MAS. Non-specific goal MAS has not any specific system goal, and all the agents in the system have themselves goals, each agent running for pursuing itself goal. There may be competitions and benefit conflicts among them. Multiagent based simulation system is a typical non-specific MAS.

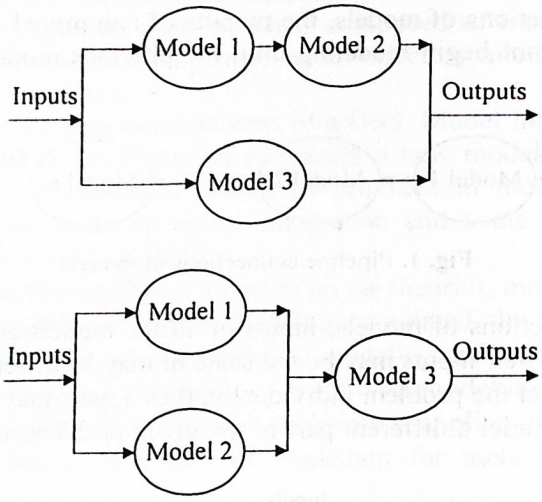


Fig. 3. Extended connections of models

Functional model integration of three modes connections can be implemented with use of cooperation of agents when we use multiagent methodology. Connections are implemented by the cooperation. We proposed a multiagent based model system which can be bused DSS and HWME. This model system can be an integrated modeling environment and a platform to support the process of model integration.

3 Multiagent Based Model System

The multiagent based model system is a specific goal system. All agents sever the same goal of providing suitable models for its users. Three kinds of working agents and one kind of support agent are designed for constructing the model system.

3.1 Components of Multiagent Based Model System

Three kinds of working agents are: management agent, interface agent and model agent.

(1) The management agent. Responsible for system management, it keeps all the information, such as the names, addresses, communication ports of agents and provides the functions of registration, updating, querying of the information for agents in the system.

(2) The interface agent. It is built for its user to interact with the model system. Every user has an interface agent. An interface agent has GUI for its user or modeler

to access data, query model information and call models. When more than one mod are used, a modeler can define the connection as pipeline, parallel or extended mod simply in graphic operating way such as selection and drag & drop.

(3) The model agent. Developed by its owner, is the main component of mod system to perform modeling and computing. Each model agent encapsulates the alg rithms of a model, which can be called by the users using their interface agents.

A multiagent based model system contains one management agent; many interfa agents for each user; many platform agents for each computer and many model agen For sake of reliability and efficiency of system, some agents may have their copi and can move to other computers.

One kind of support agent is called platform agent. It is platform for working age to be able to move from one computer to another. Each of computers in a netwo allowing the agents reside has a platform agent installed.

In multiagent based model system, all kinds of agents are cooperating for modelir in the procedure of problem solving. Cooperation of model agents can realize t model integration easily. Fig.4 shows the multiagent based model system.

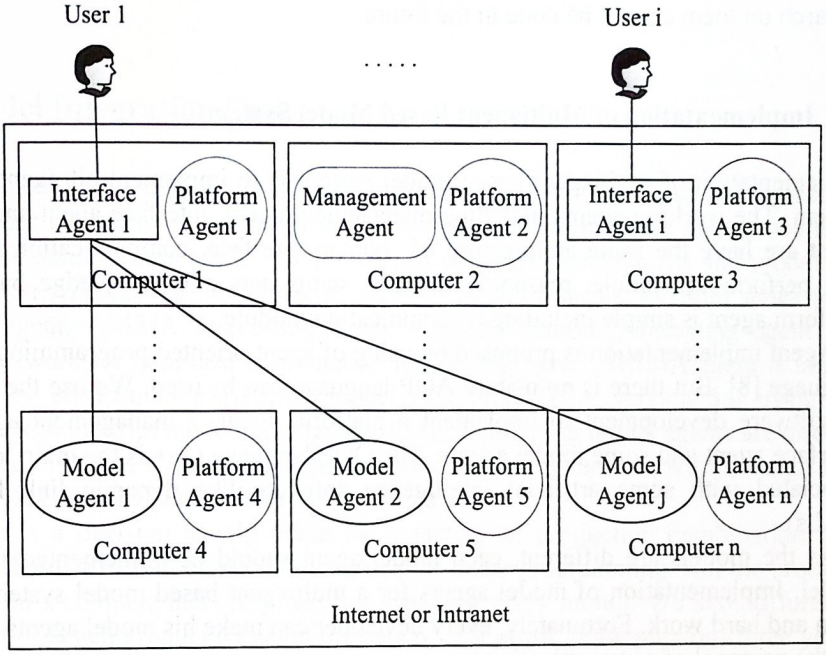


Fig. 4. Multiagent based model system

3.2 Procedure of Modeling

The procedure of modeling in manual can be described as follows:

Access or input data if it is necessary for a problem solving, such as forecasting time series forecasting, the series data in the past are absolutely necessary.

Query information of model agent in the system, and make sure that all the model agents you want to use are there in existence. One model agent may contain more

than one functions, all functions will be showed. As an ARIMA model agent, the main function is time series forecasting, and functions of calculating SAC and SPAC are still provided.

Select the model agents by dragging and dropping them to the model integration window. Specify connections of models or selected functions of model agents by drawing lines between them to define the model integration in a graphic operating interface.

Start the modeling, the interface agent will call all the required model agents to work. Modeler may be prompted with some information and be asked to input some parameters if those models require them in the modeling process.

All model agents have finished their specified work, and the final results will be sent to the interface agent of the modeler.

Interface agent displays the results for the modeler.

Automatic model integration can be done if the interface agent has the ability of problem decomposition and the model agents are capable of modeling actively. Problem decomposition rules and model integration rules are necessary for agents. Farther research on them should be done in the future.

3.3 Implementation of Multiagent Based Model System

Implementation of multiagent based model system is to implement all agents in the system. The working agents including management agent, interface agent and model agent are have the same architecture of human interface, communication module, task performing module, reasoning module, status data and knowledge base. The platform agent is simple including communication module.

Agent implementation is proposed by using of agent oriented programming (AOP) language [8]. But there is no mature AOP language can be used. We use the method of software development to implement a platform agent, a management agent, an interface agent and some model agents. The OOP language C++ is the main tool, and associated with some artificial intelligence software like dynamic link libraries (DLLs).

As the models are different, each model agent should be implemented for each model. Implementation of model agents for a multiagent based model system is the main and hard work. Fortunately, every developer can make his model agents observing the protocol of communication.

Besides the direct development of model agent by coding it with the OOP languages, model encapsulation method we proposed is another choice for implementation of model agents. We developed an agent shell (showed as Fig.5) with most capabilities of a common agent to encapsulate models of some model software, statistics software such as the STATISTICA, SAS, and linear programming software like LINDO. The calculation of modeling task is accomplished by software; other things like the communication are done by the agent shell. Unfortunately, not all the model software can be capsulated by agent shell, only those support to execute modeling descript language or provide DLLs to call are within the bounds of possibility. Though the movement of the agent with model software encapsulated is not as agile

as the directly developed model agent, the method of model encapsulation quickens the development of model agents.

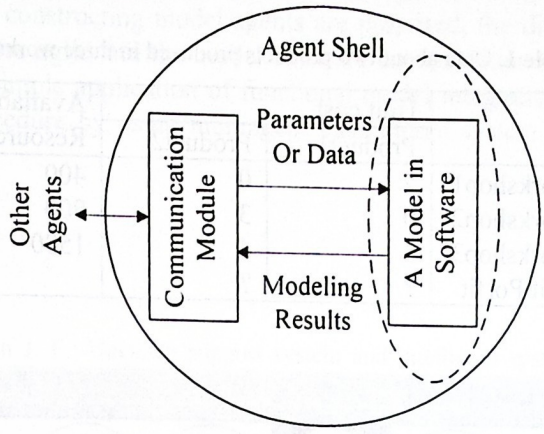


Fig. 5. Model encapsulation by agent shell

4. Model Integration Case

A multiagent based model system to support model integration has been developed by us. Besides management agent, some interface agents and platform agents, we have developed some model agents using methods of direct development and model encapsulation. Forecasting model agents includes neural network for time series forecasting agent, ARIMA forecasting agent, exponent smooth forecasting agent and neural network for nonlinear combination forecasting agent. ARIMA agent is implemented by encapsulating ARIMA model of STATSTICA version 5.0 using the STATSTICA descript modeling language. Other forecasting model agents are directly developed. There is a linear programming model agent is realized with calling functions in DLLs the LINDO version 6.1 provided.

There is a decision should make for a factory of production combination next month. That is a resource-allocation problem of linear programming. But the products are sold in a compete market, prices are changing every month. We should forecast unit profit of each product by using the data of unit profit in the previous three years. Therefore, to solve this problem, forecasting model and linear programming model must be used. This is an example of application for model integration that can demonstrate the procedure of model integration using agents in multiagent based model system.

We define the integration of models in interface agent graphically. For each of unit profit of two products, three time series forecasting models including NN time series agent, ARIMA agent, exponent smooth agent are connected in parallel, and then connect to NN combination agent. Linear programming agent is used to get the solution for problem of production combination. Fig.6 shows the model integration relationship. Unit profit forecasts of two products are 324.68 and 536.38. Then the forecasts and other data like unit cost, available resources are passed to linear program-

ming agent, final solution of this problem is gotten: 100 for product1 and 300 for product2.

Table 1. Data about two products produced in three workshops

	Unit cost		Available Resource
	Product1	Product2	
Workshop1	2	0	400
Workshop2	0	3	900
Workshop3	3	4	1500
Unit Porfit	?	?	

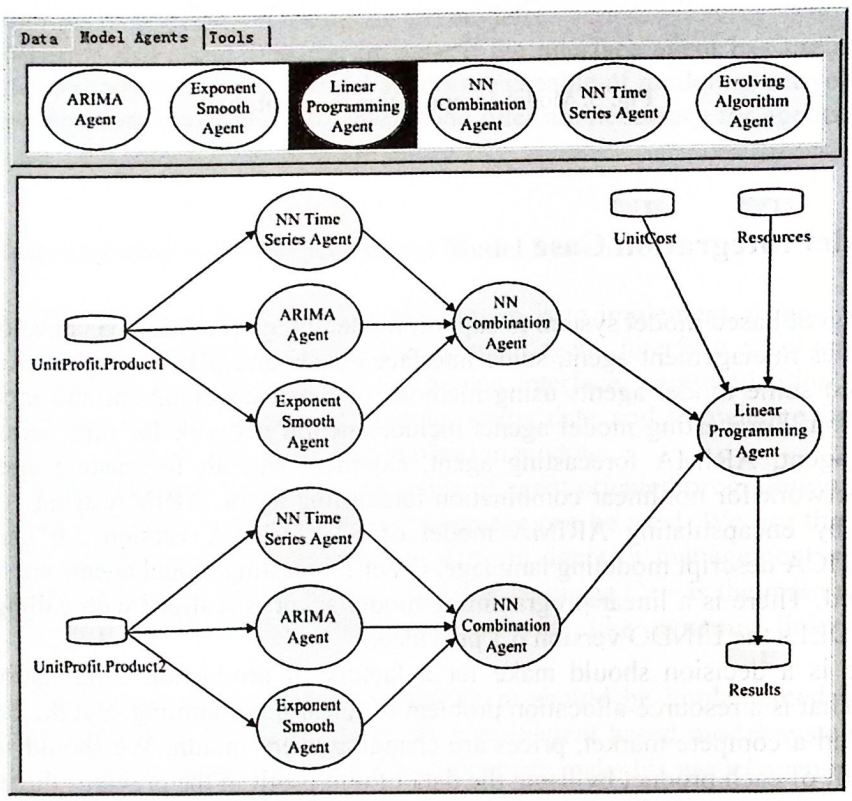


Fig. 6. Definition of the model integration

5 Conclusions

In this paper, we mainly studied the functional model integration using agents in an agent based integrated modeling platform we developed – the multiagent based model system. Connections of model are classified as two basic types: the pipeline and the

parallel, which can extend a complex linkage among models. Integration of model can be realized with the cooperation of agents in multiagent based model system.

Two methods of constructing model agents are proposed, the direct development makes the agent more agile and the model software encapsulation lets the agent developing faster. A simple application of functional model integration to demonstrate the integration procedure by using multiagent base model system we developed is presented.

References

1. Turban E., Aronson J. E.: Decision support system and intelligent system. Prentice Hall, Upper Saddle River. NJ (1999)
2. Xijin T., Jifa G. and Nakamori Y.: Model integration for system modeling. Proceedings of 2001 international symposium on knowledge and systems sciences, (2001): 285-296
3. Tung B., Jintae L.: An agent-based framework for building decision support systems. Decision Support System, 25(1999):225-237
4. Daiping H., Huanchen W., A multiagent framework for building workshop for hall of meta-synthetic engineering. Proceedings of 2001 International Symposium on Knowledge and Systems Sciences, (2001): 161-168
5. Xuesen Q., Jingyuan Y., Ruiwei D.: A new discipline of science: the study of open complex giant system and its methodology. Chinese journal of systems engineering and electronics, 4(1993): 2-12
6. Dolk D. R., Kottenmann J. E.: Model integration and a theory of models. Decision support systems, 19(1993): 51-63
7. Geoffrion A. M.: Structured modeling: Surveys and future research directions. Interactive Transactions of ORMS, 1(1996)
8. Shoham Y.: Agent-oriented Programming, Artificial Intelligence, 60 (1993): 51-92

Defeasible Decision Making in a Multi-robot Environment

Edgardo Ferretti[1], Nicolás Rotstein[2,3], Marcelo Errecalde[1], Alejandro García[2,3], and Guillermo Simari[3]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis, Argentina.
{ferretti,merreca}@unsl.edu.ar
[2] Consejo Nacional de Investigaciones Científicas y Técnicas
[3] Department of Computer Science and Engineering
Universidad Nacional del Sur, Bahía Blanca, Argentina.
{ndr,ajg,grs}@cs.uns.edu.ar

Abstract. In this work, we present a Defeasible Logic Programming approach to decision making in a multi-robot environment. It will be shown how a successful tool for knowledge representation and defeasible reasoning could be applied to the problem of deciding which task should be performed next. Besides, through several examples, we aim to show how flexible is this approach to program the robots' preference policy, considering a simple application domain with real *Khepera* 2 robots.

1 Introduction

Decision making models for autonomous agents have received increased attention, particularly in the field of intelligent robots. The proposed models are often based on formal theories of decision, such as Classical Decision Theory [1], Qualitative Decision Theory [2] and BDI logics [3]. In other cases, models from neuroscience, cognitive psychology and ethology are considered. In these models, the agents' decision making process is an emergent phenomenon of the interaction of elemental behaviors [4]. An established approach to decision making in robotic systems is that of reactive decision systems. In such systems the decision process is usually dedicated to the selection of the action to be executed, based on the current perceptual information with little (if any) pre-processing.

When applicable, the reactive approach has the advantages of simplicity and speed. However, there are domains in which reactive approaches to decision making become exceedingly difficult to apply or may not intuitively describe the behavior desired. In such cases, the agent's decision can be partly determined by immediate perceptual data but may also include a complete history of previous perceptions and decisions. The agent may also need to consider questions such as: *Which is the more appropriate goal to pursue in the current situation?* or *Which one of the alternative plans do I have to select to reach a certain goal?* Further complicating matters are that the information used in the decision process is (in most domains) incomplete and potentially inconsistent.

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 150–160

Received 16/06/07
Accepted 31/08/07
Final version 17/09/07

In this paper we will show how a Defeasible Logic Programming approach could be applied in a robotic domain for knowledge representation and reasoning about which task to perform next. At this end, we have selected a simple application domain where real robots perform cleaning tasks. We use the *Khepera* 2 robot [5], a miniature mobile robot ideal for this kind of experimentation.

The experimental environment consists of a square arena of 100 units per side which is conceptually divided into square cells of 10 units per side each. In this environment, more than one robot could be acting at the same time, but there is no communication among them. There is a global camera which provides the necessary information to perform their activities. The *store* is a 30 × 30 units square on the top-right corner and represents the target area where the boxes should be transported. There are boxes of three different sizes (*small*, *medium* and *big*) spread over the environment. At most two boxes can be stacked, but a box cannot be stacked on top of a smaller box. Thus, big boxes are always on the floor. Figure 1 shows four different scenarios of the above-described environment, depicted in a schematic way. Due to space restrictions, in each scenario, only the portion of the arena containing the involved objets is shown.

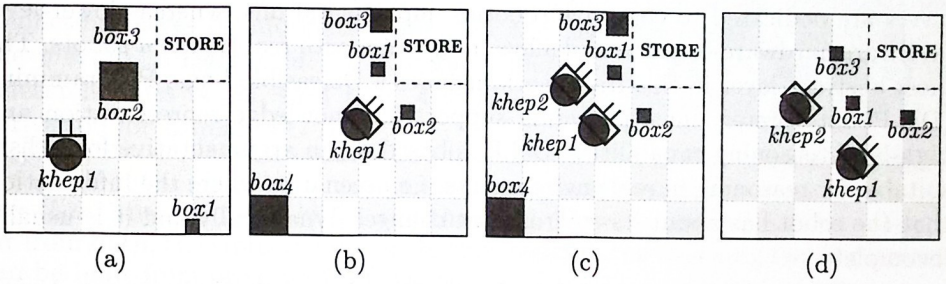


Fig. 1. Four different scenarios

Since in our proposed domain several robots cooperate and there is no communication among them, a robot will not be able to predict exactly the behavior of its partners. Therefore, the robots cannot perform a globally optimized task. In contrast, they will reason about which box is more convenient to select next. To perform the reasoning, a robot will use the perceptual information about the boxes and other robots, and its preferences which will be represented with defeasible rules. For example, a robot could prefer the smallest box, or the nearest one, or the box that is nearest to the store. As we will show below, these preferences will be defeasible and they may change according to the current situation or the presence of other robots. Arguments for and against selecting a box will be considered in order to select the more appropriate one.

A robot capable of solving this kind of problems must at least address the following issues: to perceive the surrounding world, to decide which goal has to be reached, and to have the capabilities for reaching this goal. Several architectures have been proposed in the literature which provide the agents with these skills [6–8]. In this work, we only consider the necessary reasoning processes to make

decisions about which is the most suitable box to be transported by each robot. We are not going to consider problems related to the implementation of low-level actions and sensorial perception. Moreover, some of the aspects related to the sensorial and effectorial support for the Khepera robots have received attention elsewhere [9].

The paper is organized as follows. In Sect. 2, we describe how the robots represent their knowledge and present a succinct overview of the reasoning formalism they use. Then, Sect. 3 shows how the decision process is performed through several examples, in which one or more robots are working together in the environment. Finally, Sect. 4 puts forward the conclusions and related work.

2 Khepera Robots with Argumentative Reasoning

In our approach, knowledge representation and reasoning for decision making will be done using the *Khe*-DeLP framework [9]. *Khe*-DeLP is a layered framework that provides facilities to the Khepera community for programming high-level robots' behavior using an argumentative approach. In *Khe*-DeLP, upper layers are dedicated to cognitive robotics implementations, whereas lower-level layers are hardware-oriented and allow interaction with real *Khepera* robots. The most abstract layer in this framework includes a Defeasible Logic Programming (DeLP) interpreter, thus, providing support for knowledge representation and high-level reasoning capabilities. DeLP is based on an argumentative formalism suitable for reasoning in real environments, *i.e.*, scenarios where the information that the robot has about its environment changes dynamically and it is usually incomplete.

In DeLP (see [10] for a complete description), knowledge is represented with a DeLP-program $\mathcal{P}$ that contains facts, strict rules and defeasible rules. When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules. *Facts* are ground literals representing atomic information or the negation of atomic information. In our application examples, robots' perception will be represented with facts, *e.g.*, $nearer_store(box2, box1)$.

Strict Rules are denoted $L_0 \leftarrow L_1, \ldots, L_n$ and represent firm information, whereas *Defeasible Rules* are denoted $L_0 \prec L_1, \ldots, L_n$ and represent tentative information. In both cases, the *head* L_0 is a literal and the *body* $\{L_i\}_{i>0}$ is a set of literals. As usual in *Logic Programming*, variables are denoted with an initial uppercase letter. Figure 2 shows the DeLP-program that will be used in our application examples. There, for example, the strict rule "$choose(R, B) \leftarrow unique(B)$" represents that "if B is the only box in the environment, then it must be chosen by robot R", and "$\sim choose(R, B) \leftarrow on(Obox, B)$" states that "if B has a box $Obox$ on its top then do not choose B". In the same figure, the defeasible rule "$choose(R, B) \prec better(R, B, Obox)$" states that " if box B is better than box $Obox$, then robot R has a (defeasible) reason for choosing B", whereas "$\sim choose(R, B) \prec better(R, Obox, B)$" states that "if there is other box that is better than B, then there is a reason for not choosing B". Observe that strong

negation ("$\sim$") is allowed in the head of program rules, and hence may be used to represent contradictory knowledge.

$$better(R, B, Obox) \prec nearer_robot(R, B, Obox) \quad (1)$$
$$better(R, B, Obox) \prec nearest_robot(R, B) \quad (2)$$
$$better(R, B, Obox) \prec smaller(B, Obox) \quad (3)$$
$$better(R, B, Obox) \prec nearer_store(B, Obox) \quad (4)$$
$$\sim better(R, B, Obox) \prec nearer_robot(R, Obox, B) \quad (5)$$
$$\sim better(R, B, Obox) \prec nearest_robot(R, Obox), diff(B, Obox) \quad (6)$$
$$\sim better(R, B, Obox) \prec nearer_store(Obox, B) \quad (7)$$
$$\sim better(R, B, Obox) \prec smaller(Obox, B) \quad (8)$$
$$\sim better(R, B, Obox) \leftarrow same_properties(R, Obox, B) \quad (9)$$
$$\sim better(R, Obox, B) \leftarrow same_properties(R, Obox, B) \quad (10)$$
$$choose(R, B) \prec better(R, B, Obox) \quad (11)$$
$$\sim choose(R, B) \prec better(R, Obox, B) \quad (12)$$
$$\sim choose(R, B) \prec better(R, B, Obox), better(R, BetterB, B), diff(Obox, BetterB) \quad (13)$$
$$choose(R, B) \prec same_properties(R, Obox, B) \quad (14)$$
$$\sim choose(R, B) \prec diff_robot(R, Or), better(Or, B, _), choose(R, Obox), diff(B, Obox) \quad (15)$$
$$choose(R, B) \leftarrow unique(B) \quad (16)$$
$$\sim choose(R, B) \leftarrow on(Obox, B) \quad (17)$$
$$\sim choose(R, B) \leftarrow carrying(Or, B) \quad (18)$$

Fig. 2. $\mathcal{P}_k = (\Pi_k, \Delta_k)$

In DeLP, to deal with contradictory and dynamic information, *arguments* for conflicting pieces of information are built and then compared to decide which one *prevails*. An *argument* for a literal L, denoted $\langle \mathcal{A}, L \rangle$, is a minimal set of defeasible rules $\mathcal{A} \subseteq \Delta$, such that $\mathcal{A} \cup \Pi$ is non-contradictory and there is a derivation for L from $\mathcal{A} \cup \Pi$.

For instance, let us consider the situation depicted in Fig. 1(a), where there is a single robot ($khep1$), a medium size box ($box3$) near to the store, and a big box ($box2$) near to both $khep1$ and the store. There is also a small box ($box1$) far from both, the store and $khep1$. In this scenario, the following two arguments can be built from program $\mathcal{P}_k$ (Fig. 2):

$$\mathcal{A}_1 = \left\{ \begin{array}{l} choose(khep1, box2) \prec better(khep1, box2, box3) \\ better(khep1, box2, box3) \prec nearer\ robot(khep1, box2, box3) \end{array} \right\}$$

$$\mathcal{A}_2 = \left\{ \begin{array}{l} \sim choose(khep1, box2) \prec better(khep1, box3, box2) \\ better(khep1, box3, box2) \prec smaller(box3, box2) \end{array} \right\}$$

In DeLP, to establish if $\langle \mathcal{A}, L \rangle$ is a non-defeated argument, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\langle \mathcal{A}, L \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, L \rangle$. For example, $\mathcal{A}_1$ is a counter-argument for $\mathcal{A}_2$, and *vice versa*. Since counter-arguments are arguments, defeaters for them may exist, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor in the line (for a detailed explanation of this dialectical process see [10]). The prevailing argument provides a warrant for the information it supports. In DeLP, a literal L is *warranted* from (Π, Δ) if a non-defeated argument $\mathcal{A}$ supporting L exists. For instance, $\mathcal{A}_1$ and $\mathcal{A}_2$ are two conflicting arguments and, as will be explained next, $\mathcal{A}_1$ is a proper defeater for $\mathcal{A}_2$, thus warranting the literal $choose(khep1, box2)$.

In DeLP, the comparison criterion among arguments can be established in a modular way. Thus, an appropriate domain-dependent criterion can be used. The criterion that we will use in our approach is defined next.

Definition 1 (L-order). *Let $\mathcal{P}$ be a DeLP program and lits($\mathcal{P}$), the set of literals in $\mathcal{P}$. An* L-order *over $\mathcal{P}$ is a partial order over the elements of lits($\mathcal{P}$).*

Regarding physical aspects of the robots, their autonomy is limited, and they cannot measure the state of their batteries. Because of this drawback, a *greedy* strategy is used to select the next box. Therefore, the robot will *prefer its nearer boxes, then the boxes nearer to the store, and finally the smaller ones.* These preferences will be explicitly established using a preference order among the literals *smaller*, *nearer_store*, *nearer_robot* and *nearest_robot*. As will be shown below, an L-order must be provided as a set of facts within the program. These facts are written as $X > Y$, stating that a literal X is preferred to a literal Y, and they will be used to decide when an argument is better than another. In particular, the L-order defined in Fig. 3 represents the robots preferences over boxes, *i.e.*, it shows the L-order defined over program $\mathcal{P}_k$ (Fig. 2), used in our application examples. Based on a given L-order, the following argument comparison criterion can be defined.

Definition 2. *Let $\mathcal{P} = (\Pi, \Delta)$ be a DeLP-program and let "$<$" be an L-order over $\mathcal{P}$. Given two argument structures $\langle \mathcal{A}_1, h_1 \rangle$ and $\langle \mathcal{A}_2, h_2 \rangle$, the argument $\langle \mathcal{A}_1, h_1 \rangle$ will be preferred over $\langle \mathcal{A}_2, h_2 \rangle$ iff:*
1. there are two literals L_1 and L_2 such that $L_1 \in^* \mathcal{A}_1$, $L_2 \in^* \mathcal{A}_2$, $L_2 < L_1$, and
2. there are no literals L_1' and L_2' such that $L_1' \in^* \mathcal{A}_1$, $L_2' \in^* \mathcal{A}_2$, and $L_1' < L_2'$.
where $L \in^* \mathcal{A}$ iff there exists a defeasible rule $(L_0 \prec L_1, L_2, \ldots, L_n)$ in $\mathcal{A}$ and $L = L_i$ for some i $(0 \leq i \leq n)$.

On the basis of this criterion, defeaters can be either proper or blocking. Given an argument comparison criterion C, an argument $\mathcal{A}_1$, and a counter-argument $\mathcal{A}_2$ for it: $\mathcal{A}_2$ is a proper defeater for $\mathcal{A}_1$ iff $\mathcal{A}_2$ is better than $\mathcal{A}_1$ wrt. C; $\mathcal{A}_2$ is a blocking defeater for $\mathcal{A}_1$ iff neither of both is better than the other wrt. C; finally, $\mathcal{A}_2$ is not a defeater for $\mathcal{A}_1$ iff $\mathcal{A}_1$ is better than $\mathcal{A}_2$ wrt. C.

In DeLP, a query Q has four possible answers: YES, if Q is warranted; NO, if the complement of Q is warranted; UNDECIDED, if neither Q nor its complement are warranted; and UNKNOWN, if Q is not in the signature of the program.

$nearest_robot(R,Z) > nearer_robot(R,X,Y)$	(19)	$nearer_store(Z,X) > smaller(X,Y)$	(24)
$nearer_robot(R,Z,X) > nearer_store(X,Y)$	(20)	$nearer_robot(R,Z,X) > smaller(X,Y)$	(25)
$nearer_robot(R,Z,Y) > smaller(X,Z)$	(21)	$nearer_store(Z,X) > smaller(Y,Z)$	(26)
$nearer_robot(R,Z,Y) > nearer_store(X,Z)$	(22)	$nearest_robot(R,Z) > smaller(X,Y)$	(27)
$nearest_robot(R,Z) > nearer_store(X,Y)$	(23)	$smaller(Z,X) > smaller(X,Y)$	(28)

Fig. 3. L-order over lits($\mathcal{P}_k$)

3 Selecting Boxes through Argumentation

In classical decision making domains it is usually assumed that the agent's choice behavior is modeled with a binary *preference relation* $\succsim$ where, $x \succsim y$ means that "x is at least as good as y". From $\succsim$ we can derive the *strict preference* relation

$\succ$[4] and the *indifference* relation $\sim$.[5] It is common to require the preference relation $\succsim$ to be *rational* (complete and transitive) and this is a necessary condition if $\succsim$ will be represented by a utility function. According to this *preference-based approach* (PBA), we should prove that the DeLP-program of Fig. 2 and the argumentation process involved when comparing two boxes, implement an implicit rational preference relation between any pair of alternatives (boxes) that could be presented to the robot. Despite of the relevance of the PBA from a theoretical point of view, an analysis of this type is beyond the scope of this paper. Besides, it is not always guaranteed that this approach accurately reflects the decision making capabilities of agents facing decision problems of the real world.[6]

In our work, we attempt to show how flexible can be an approach based on defeasible argumentation when new and changing information has to be considered during the decision making process. For this reason, the *static* properties of the preferences of our robots are not so important, and it should be useful to directly consider the *choice behavior* of the robots and to analyze the *dynamics* of the decision processes, when new alternatives have to be considered due to changes in the environment or when new information is available to the robot. From this point of view, an interesting and more flexible formal model of theory of decision making, called *choice-based approach* [11] provides us more adequate tools for evaluating the dynamics involved in the decisions of the robot.

The *choice-based approach* (CBA) takes the individual's choice behavior as a primitive object, which is represented by means of a *choice structure* $(\mathcal{B}, C(\cdot))$ where $\mathcal{B}$ is a family (a set) of subsets of X (the set of possible alternatives). Intuitively, each set $B \in \mathcal{B}$ (where $B \subset X$) represents each set of alternatives (or *choice experiments*) that can be conceivably posed to the decision maker. In this way, if $X = \{x, y, z\}$ and $\mathcal{B} = \{\{x, y\}, \{x, y, z\}\}$, we will assume that the sets $\{x, y\}$ and $\{x, y, z\}$ are valid choice experiments to be presented to the decision maker. $C(\cdot)$ is a *choice rule* (a correspondence) which basically assigns to each set of alternatives $B \in \mathcal{B}$ a non-empty set that represents the alternatives that the decision maker *might* choose when presented the alternatives in B. We can note that $C(B) \subset B$ for every $B \in \mathcal{B}$ and when $C(B)$ contains a single element, this element represents the *individual's choice* from among the alternatives in B. The $C(B)$ set may, however, contain more than one element and in this case they represent the *acceptable alternatives* in B for the agent.

A central assumption in this approach, the *weak axiom of revealed preference* (WARP), imposes an element of consistency on choice behavior in a sense paralleling the rationality assumptions of the PBA. Intuitively, the WARP principle reflects the expectation that an individual's observed choices will display a certain amount of consistency. For example, if an individual chooses alternative x (and only that) when facing the alternatives $\{x, y\}$, we would be surprised to

[4] Defined as $x \succ y \Leftrightarrow x \succsim y$ but not $y \succsim x$ and read "x is preferred to y"

[5] Defined as $x \sim y \Leftrightarrow x \succsim y$ and $y \succsim x$ and read "x is indifferent to y"

[6] It is well known in the decision theory community that completeness and transitivity assumptions are usually very hard to satisfy in real world agents when evaluating alternatives far from common experience [11].

see him choose y when faced the alternatives $\{x, y, z\}$. More formally, the weak axiom postulates that "if there is some choice experiment $B \in \mathcal{B}$ such that x and y are presented as alternatives $(x, y \in B)$ and "x is *revealed at least as good as* y $(x \in C(B))$ then it does not exist other choice experiment $B' \in \mathcal{B}$ where "y is *revealed preferred* to x $(x, y \in B', y \in C(B')$ and $x \notin C(B'))$.

In our work, the robots are essentially facing a discrete multi-attribute decision problem, where each alternative (box) would represent a complete assignment of attribute values. The set X of alternatives available to the robots would be represented by the properties associated to the boxes to be considered in the decision process. For this reason, we will use the notation x_{box_i} to denote the description of the attributes corresponding to box box_i, *e.g.*, "near to the robot," "near to the store," etc.

It is direct to note the similarities of our approach with the scenario assumed in the CBA; where the *choose*/2 predicate (Fig. 2) would play the role of the choice rule $C(\cdot)$. In this case, if n boxes $box_1, \ldots, box_n$ are present in the environment, the robot faces a choice experiment with n alternatives $\{x_{box_1}, \ldots, x_{box_n}\}$ with the properties corresponding to each box present in the environment. In this context, when the DeLP answer to the query $choose(R, box_k)$ is YES, it should be interpreted as saying that the properties of box_k make it an acceptable alternative for the robot R, or more formally, the robot R has a choice rule $C(\cdot)$ such that $x_{box_j} \in C(\{x_{box_1}, \ldots, x_{box_n}\})$. From this perspective, is direct that the WARP principle can be easily restated in terms of the *choose* predicate. We can say in this case, that *choose*/2 satisfies the weak axiom of revealed preference if every time that two boxes with attributes x_j and x_k are considered and *choose* answers YES to the box with attributes x_j, it does not exist another situation where two boxes with the same attributes are considered and the answer of *choose* for x_k is YES and it does not respond YES for the box x_j.

Below, we show through several examples how the robots react in different situations that might arise in our dynamic application domain. Besides, it will be pointed out how the robots decisions respect the WARP principle.

Example 1. Let us consider first a simple situation (Fig. 1(a)) with one robot (*khep*1) and three boxes: *box*1 (small), *box*2 (big) and *box*3 (medium). Here, *box*2 and *box*3 are near to the store, *box*2 is near to the robot, and *box*1 is far from both, robot and store. Considering *khep*1*'s* preferences (Fig. 3) *box*2 should be chosen because, despite of its size, is the only box near to the robot and also to the store. The robot's perception with regards to this situation is the set:

$$\Pi_a = \left\{ \begin{array}{l} smaller(box1, box2),\ nearer_store(box2, box1),\ nearer_robot(khep1, box2, box3), \\ smaller(box1, box3),\ nearer_store(box3, box1),\ nearer_robot(khep1, box2, box1), \\ smaller(box3, box2) \end{array} \right\}$$

The DeLP-program of *khep*1 includes $\mathcal{P}_k$, Π_a and the L-order of Fig. 3. From this program, DeLP builds the following undefeated arguments:

$$\mathcal{A}_{1.1} = \left\{ \begin{array}{l} \sim choose(khep1, box1) \prec better(khep1, box2, box1) \\ better(khep1, box2, box1) \prec nearer_store(box2, box1) \end{array} \right\}$$

$$\mathcal{A}_{1.2} = \left\{ \begin{array}{l} choose(khep1, box2) \prec better(khep1, box2, box1) \\ better(khep1, box2, box1) \prec nearer_store(box2, box1) \end{array} \right\}$$

$$\mathcal{A}_{1..3} = \left\{ \begin{array}{l} \sim choose(khep1, box3) \prec better(khep1, box2, box3) \\ better(khep1, box2, box3) \prec nearer_robot(khep1, box2, box3) \end{array} \right\}$$

Hence, as expected from the preferences encoded in Fig. 3, the DeLP answer for $choose(khep1, box1)$ is NO, for $choose(khep1, box2)$ is YES, and for $choose(khep1, box3)$ is NO.

Example 2. In order to illustrate how changes in a dynamic environment are handled by the DeLP-program $\mathcal{P}_k$, consider that in the scenario depicted in Fig. 1(a) a new medium size box ($box4$) is placed on top of $box2$. It is clear that $box2$ is no longer the best choice, and the robot should select $box4$. In this new scenario, the fact $on(box4, box2)$ is added, and rule (17) blocks any argument supporting $choose(khep1, box2)$. Hence, the DeLP answer for $choose(khep1, box2)$ is NO, whereas $choose(khep1, box4)$ returns YES.

Example 3. Consider the situation of Fig. 1(b), with one robot ($khep1$) and four boxes: $box1$ (small), $box2$ (small), $box3$ (medium) and $box4$ (big). Here, $box1$ and $box2$ are near to the robot and to the store, $box3$ is far from the robot and near to the store, and $box4$ is far from both, robot and store. The robot's perception is:

$$\Pi_b = \left\{ \begin{array}{ll} smaller(box1, box3), & smaller(box1, box4), \\ smaller(box2, box3), & smaller(box2, box4), \\ smaller(box3, box4), & nearer_store(box1, box4), \\ nearer_store(box2, box4), & nearer_store(box3, box4), \\ nearest_robot(khep1, box1), & nearest_robot(khep1, box2), \\ nearer_robot(khep1, box1, box3), & nearer_robot(khep1, box2, box3), \\ nearer_robot(khep1, box1, box4), & nearer_robot(khep1, box2, box4), \\ same_properties(khep1, box1, box2) & \end{array} \right\}$$

It is clear that $box1$ and $box2$ are better alternatives than $box3$ and $box4$, hence, the latter ones should not be chosen. Nonetheless, as $box1$ and $box2$ have the same properties (from $khep1$'s point of view) these alternatives should be indifferent to $khep1$. The DeLP-program of $khep1$ will include $\mathcal{P}_k$, Π_b and the L-order of Fig. 3. From this program, the answer for both $choose(khep1, box1)$ and $choose(khep1, box2)$ will be YES, representing that the robot presents equal preference (*i.e.*, indifference) to both boxes. As it was expected, the answers for $choose(khep1, box3)$ and $choose(khep1, box4)$ will be NO.

Example 4. Taking into account the WARP principle mentioned at the beginning of this section, the choices made by $khep1$ in Ex. 3 should not be changed if $box3$ and $box4$ were removed from the scenario. This is because these boxes (alternatives) are worse than $box1$ and $box2$, which have the same properties independently of the presence or absence of other boxes. If we consider a scenario that only includes $box1$ and $box2$, the robot's perception would be:

$$\Pi_{b'} = \left\{ \begin{array}{ll} nearest_robot(khep1, box1), & nearest_robot(khep1, box2), \\ same_properties(khep1, box1, box2) & \end{array} \right\}$$

From the DeLP-program including $\mathcal{P}_K$, $\Pi_{b'}$ and the L-order of Fig. 3, as expected, the answer to $choose(khep1, box1)$ and $choose(khep1, box2)$ is YES.

Example 5. As shown in Ex. 4, some decisions should be maintained if the changes in the world are not significant to the robot's decision-making policy.

Nonetheless, if we consider the situation shown in Fig. 1(c), where a new robot ($khep2$) appears in the environment, this fact might modify $khep1's$ choice depending on $khep2's$ position. Now, $box1$ is the nearest box of $khep2$ and from its point of view $box1$ is its best choice. In face of this new evidence, $khep1$ still have a nearest box ($box2$) to choose instead of $box1$. Since the only difference from the scenario depicted in Fig. 1(b) is the presence of $khep2$, the perception now is $\Pi_c = \Pi_b \cup \Phi$, where:

$$\Phi = \left\{ \begin{array}{ll} nearest_robot(khep2, box1), & nearer_robot(khep2, box1, box2), \\ nearer_robot(khep2, box1, box3), & nearer_robot(khep2, box1, box4) \end{array} \right\}$$

At first sight, the alternatives (boxes) available to $khep1$ seem to be the same that in Ex. 3. Nevertheless, with the presence of $khep2$ in the environment, the alternative associated with $box1$ changes because, as represented in Φ, $box1$ has the new attribute of being the nearest box of another robot ($khep2$).

Note that rule (15) plays a fundamental role, because it allows the robots to choose a certain box, considering the other robots' preferences. Without this rule $khep1$ would have chosen $box1$ and $box2$ as in Ex. 3. Then, a problematic situation can arise if both robots intend to grab the same box.

Since both robots have the same preferences and the same perception (Π_c), their DeLP-program will include $\mathcal{P}_k$, Π_c and the L-order of Fig. 3. From this program, the answers for $choose(khep1, box3)$ and $choose(khep1, box4)$ remains being NO, whereas the answer for $choose(khep1, box2)$ remains being YES, since the presence of $khep2$ does not compromise $khep1$'s decision. In opposition, the query $choose(khep1, box1)$ is now UNDECIDED, because it is a box that could be selected by $khep2$. Regarding $khep2$, the DeLP answer for $choose(khep2, box1)$ will be YES, and for $choose(khep2, box2)$, $choose(khep2, box3)$ and $choose(khep2, box4)$ will be NO, since they do not represent good choices for this robot.

Example 6. Let us consider a final example regarding the situation depicted in Fig. 1(d). As can be observed, there are three small boxes ($box1$, $box2$ and $box3$) and two robots ($khep1$ and $khep2$). All the boxes are near to the store and in particular $box1$ is near to both robots. Then, $box2$ is near to $khep1$ and far from $khep2$ while $box3$ is near to $khep2$ and far from $khep1$. The boxes $box1$ and $box2$ have the same properties from $khep1's$ standpoint, the same occurs with $box1$ and $box3$ from $khep2's$ point of view. Since both robots can choose other boxes besides $box1$, and they are as good as $box1$, intuitively the best combination would be $khep1$ choosing $box2$ with $khep2$ choosing $box3$. The robots' perception in this situation is the following set:

$$\Pi_d = \left\{ \begin{array}{ll} nearest_robot(khep1, box1), & nearest_robot(khep1, box2), \\ nearest_robot(khep2, box1), & nearest_robot(khep2, box3), \\ nearer_robot(khep1, box2, box3), & nearer_robot(khep2, box3, box2), \\ same_properties(khep1, box1, box2), & same_properties(khep2, box1, box3) \end{array} \right\}$$

From the DeLP-program that includes $\mathcal{P}_k$, Π_d and the L-order of Fig. 3, as expected, the answers for $choose(khep1, box2)$ and $choose(khep2, box3)$ are YES, whereas the answer for $choose(khep1, box3)$ and $choose(khep2, box2)$ are NO. It is important to note that the answer for both queries $choose(khep1, box1)$ and $choose(khep2, box1)$ is UNDECIDED. The reason why this occurs is that DeLP cannot build an undefeated argument to support neither $choose(khep1, box1)$ nor $\sim choose(khep1, box1)$, and the same situation holds for $khep2$.

Changing Preferences

To sum up, it worths noticing that this defeasible approach to decision making is flexible enough to easily modify preferences, and thus obtain a different behavior from the robot. As stated in Sect. 2, in the above examples a robot prefers its nearer boxes, then the boxes nearer to the store, and finally the smaller ones. These preferences were explicitly established by the L-order given in Fig. 3, but they can be replaced in a modular way. For instance, another possibility would be giving highest preference to the smallest boxes, *i.e.*, the size of each box would be more important than its proximity to the store or to the robot. This would imply to change in Fig. 3 the facts (21), (24), (25), and (26) for the following ones: $smaller(Z, Y) > nearer_store(X, Z)$, $smaller(Z, X) > nearer_robot(R, Y, Z)$ and $smaller(X, Y) > nearest_robot(R, Y)$. Taking into account these changes, let us reconsider Ex. 1. Now, the DeLP answers to the queries $choose(khep1, box1)$, $choose(khep1, box2)$, and $choose(khep1, box3)$ will be YES, NO, and NO, respectively. Thus, according to this new L-order, $khep1$ now chooses $box1$ (the smallest one) instead of $box2$ (the nearest one).

4 Conclusions and Related Work

In this paper we have shown how a Defeasible Logic Programming approach could be applied in a robotic domain for knowledge representation and reasoning about which task to perform next. Our approach considers the ability of Defeasible Logic Programming to reason with incomplete and potentially inconsistent information. Through several examples we aimed to show the flexibility of this approach to program the robots' preference policy, by considering a simple application domain where real *Khepera* 2 robots perform cleaning tasks.

In particular, in our work we follow some ideas exposed by Parsons *et al.* [12] about the integration of high-level reasoning facilities with low-level robust robot control. We share the approach of seeing the low-level module as a black box which receives goals to be achieved from the high-level component, and plans to reach goals are internally generated. However, our work differs from [12] in that we do not use a BDI deliberator as high-level reasoning layer, instead we use a non-monotonic reasoning module based on a defeasible argumentation system.

With respect to this last issue, our approach to decision making is related to other works which use argumentative processes as a fundamental component in the decision making of an agent [13–15]. It is important to note that these argumentation systems have been usually integrated in *software* agents. On the other hand, in our approach, defeasible argumentation is applied in a robotic domain where uncertainty (generated by noisy sensors and effectors), changes in the physical environment, and incomplete information about it, make this kind of problems a more challenging test-bed for the decision processes of an agent.

Finally, since Prolog is a programming language that has already been used to develop applications in the field of cognitive robotics [16], we will make some final brief comments on the differences that arise when programming a cognitive agent in Prolog with respect to DeLP. A comparison of this nature can be rather unfair, since DeLP was intended from the beginning to be a high-level

logic programming language for knowledge representation and reasoning. In our approach, robots' preferences are modeled as a partial order among literals. Therefore, in DeLP, when these preferences are changed, it will only affect the set of facts representing them. That is because the DeLP formalism allows for a more modular way to represent knowledge. In opposition to this, in Prolog, changing the preferences may involve checking and rewriting most of the program. Hence, the developer would have to specify each and every scenario within which a robot makes a certain choice.

Acknowledgment

We thank the Universidad Nacional de San Luis and the Universidad Nacional del Sur for their unstinting support. This work is partially supported by, CONICET (PIP 5050), and ANPCyT (PICT 2002, Nro.13096 and Nro.12600).

References

1. Jeffrey, R.C.: The Logic of Decision. 2nd edn. University Of Chicago Press (1990)
2. Doyle, J., Thomason, R.H.: Background to qualitative decision theory. AI Magazine (1999)
3. Rao, A., Georgeff, M.: Decision procedures for BDI logics. Journal of Logic and Computation (1998)
4. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation (1986)
5. K-Team: Khepera 2. http://www.k-team.com (2002)
6. Gat, E.: On three-layer architectures. In: Artificial Intelligence and Mobile Robots. (1998)
7. Estlin, T., Volpe, R., Nesnas, I., Muts, D., Fisher, F., Engelhardt, B., Chien, S.: Decision-making in a robotic architecture for autonomy. In: International Symposium, on AI, Robotics and Automation for Space. (2001)
8. Rotstein, N., García, A., Simari, G.: Reasoning from desires to intentions: A dialectical framework. In: Proceedings of the 22nd. AAAI Conference on Artificial Intelligence. (2007) 136–141
9. Ferretti, E., Errecalde, M., García, A., Simari, G.: KheDeLP: A framework to support defeasible logic programming for the khepera robots. In: ISRA. (2006)
10. García, A., Simari, G.: Defeasible logic programming: An argumentative approach. Theory Practice of Logic Programming 4(1) (2004) 95–138
11. Mas-Collel, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press (1995)
12. Parsons, S., Pettersson, O., Saffiotti, A., Wooldridge, M.: Robots with the Best of Intentions. In: Artificial Intelligence Today: Recent Trends and Developments. Springer (1999)
13. Atkinson, K., Bench-Capon, T.J.M., Modgil, S.: Argumentation for decision support. In: DEXA. (2006) 822–831
14. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: AAMAS. (2003)
15. Parsons, S., Fox, J.: Argumentation and decision making: A position paper. In: FAPR. (1996) .
16. Levesque, H.J., Pagnucco, M.: Legolog: Inexpensive experiments in cognitive robotics. In: 2nd International Cognitive Robotics Workshop. (2000)

Normative Conflict Detection for e-Contracts[1]

Georgios K. Giannikis & Aspassia Daskalopulu

Department of Computer and Communications Engineering,
University of Thessaly, Gklavani 37, 38221 Volos, Greece
{ggiannik,aspassia}@inf.uth.gr

Abstract. This paper is concerned with normative conflicts that arise for agents engaging in electronic contracting, and presents a set of primitive conflict patterns. We examine other analyses of normative conflicts and show how these may be seen as instances of these primitives. We also identify some patterns of normative conflict that have not been identified in other proposals. Finally, we propose a representation of e-contracts in Default Logic, which facilitates the detection and resolution of such normative conflicts.

Keywords: Conflict Patterns, Conflict Detection, E-contracts, Default Logic

1 Introduction

The analysis, representation and management of normative conflicts has been the focus of much research in recent years, from a variety of perspectives, such as distributed systems management, legal reasoning and argumentation, and multi-agent interaction.

In the early nineties, Sartor's [25] and Horty's [12] work on normative conflicts set the theoretical basis for conflict management. According to Sartor [25] a conflict arises when "(possibly) valid norms establish incompatible qualifications for the same concrete state". The corner stone in this approach is a *norm* set. This may be either *inconsistent*, if a contradiction is logically derivable from it, or *potentially inconsistent*, if it may lead to contradiction in an upcoming state. In similar spirit Horty in [12, 13] addresses moral conflicts: an agent is in moral conflict if it ought to do an action A and, at the same time, it ought to do another action B, but it is impossible to do both. Moffett *et al.* [21], Lupu *et al.* [18] and Dunlop *et al.* [7, 8] address conflicts from the Distributed Systems Management viewpoint and view policies as a way to determine and influence management behaviour. Cholvy *et al.* [4, 5] view normative conflicts as the result of role conflict and propose a solution based on hierarchies of roles. Broersen *et al.* [3] deal with conflicts that arise between an agent's mental attributes such as beliefs, obligations, intensions and desires. Kowalski [16] considers normative conflicts that arise as a result of conflicting goals and presents an approach

[1] This work was supported by the European Commission and the Greek Secretariat for Research and Technology (PENED 2003 – 03EΔ466).

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 161–172

Received 16/06/07
Accepted 31/08/07
Final version 16/09/07

that unifies logic and decision theory. Finally, Kollingbaum *et al.* [15] focus on practical reasoning agents and use instantiation graphs to detect conflicts.

In this work we are concerned with normative conflicts that arise for agents engaging in electronic contracting, within an electronic marketplace, and we investigate an alternative representation, in which we use Reiter's Default Logic (DfL) [23]. In [9, 10] we proposed the representation of contractual norms as default rules, which are constructed dynamically from temporal representations. The resulting default theories afford us both temporal and defeasible reasoning, as well as facilitating conflict resolution [9]. Here, we identify a set of primitive patterns for normative conflicts and show how the conflicts identified by other researchers may be seen as instances of these primitives. We also identify some patterns of normative conflict that have not been identified in other proposals.

2 Conflict Detection

For the purposes of illustration consider an electronic marketplace, populated by software agents that establish and perform e-contracts on behalf of some real world parties. Let the set *Agents={Agent1, Agent2, Agent3,.....}* denote distinct identifiers for the various agents, and the set *Roles={RA, WA, MA, CA, ...}* denote distinct roles that agents may assume in the e-market (where *RA*, *WA*, *MA*, *CA* denote *retailer*, *wholesaler*, *mediator* and *carrier* respectively).

Consider a two-party business transaction. *Agent1* that acts as a retailer orders some goods from the wholesaler *Agent3*. The terms of the agreement between these two agents are: *Agent3* should see to it that the goods be delivered to *Agent1* within 10 days from commencement (e.g., the date that the order takes place). *Agent1*, in turn, should see to it that payment be made within 21 days from the date it receives the goods. If *Agent3* does not deliver on time, then a fixed amount is to be deducted from the original price of the goods for each day of delay and it should see to it that delivery be made by a new deadline. If *Agent1* does not perform payment on time, then a fixed amount is to be added to the original price of the goods for each day of delay and it should see to it that payment be made by a new deadline.

Following [6], we may take an informal, process view of the business transaction that is regulated by the agreement. Each state offers a (possibly partial) description of the factual and normative propositions that hold true. A transition between states corresponds to an event that takes place, i.e. an action that one of the parties performs or omits to perform. Normative propositions of the form:

NN(agent1, role1, action, agent2, role2)

express that *agent1* that acts as *role1* is in legal relation *NN* towards *agent2* that acts as *role2* to perform *action*, where *NN* may be *Obligation*, *Prohibition*, *Permission* and legal *Power*.

We use Reiter's Default Logic [23] to represent the norms of an agreement as default rules. A default rule has the form *P:J1,J2,...Jn/C,* where *P* is the *prerequisite*, *J={J1,J2,... Jn}* is a set of *justifications* and *C* is the derived *consequent*. If *J* coincides with *C*, the default rule is called *normal*. The semantics of a default rule is: If *P* holds and the assumption *J* is consistent with our current knowledge, then *C* may be inferred.

For instance, the following default rule expresses that if an order from *Agent1* (acting as a retailer) towards *Agent3* (acting as a wholesaler) holds, and it is consistent to assume that *Agent1* will become a regular client, then we may infer that *Agent3* is legally obliged towards *Agent1* to perform delivery:

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

A Default Theory (DfT) is a pair of the form (W, D), where W is a set of closed formulae that hold, and D is a set of defaults. Rules may be used to compute *extensions* E of the default theory. A rule is applicable to a set of formulae $W \subseteq E$ if and only if $P \in E$ and $\neg J_1, \ldots, \neg J_n \notin E$. We consider grounded DfTs and we derive extensions in the manner presented in [2], i.e. by maintaining consistent sets of formulae. This derivation may be conducted in stepwise manner. Thus, an agent that engages in a transaction governed by some agreement, essentially reasons with a default theory. At each time point during the business transaction the agent attempts to compute the extensions of its current DfT. Note that whenever multiple extensions are computed for a Default Theory these represent possible world views. Depending on its chosen action an agent is committed to a particular extension. The DfT contract representation allows us to detect normative conflicts by examining extensions. A normative conflict may be detected either between multiple extensions or between some extension and the current knowledge (W) of the agent. Where a conflict is detected between multiple extensions, the latter represent alternative futures for the agent; let us call these *inter-extension* conflicts. Where a conflict is detected between an extension and the current knowledge of the agent, it represents a state in which some normative violation will eventually occur; let us call these *intra-extension* conflicts. The role of conflict detection is, thus, to assist an agent to choose a course of action so that normative violations may be predicted and avoided.

The first step of conflict management involves the detection of conflicts. To this end, in section 2.1, we identify primitive patterns of normative conflict that may be spotted during the derivation of extensions of the default theory representation of an agreement. In section 2.2 we discuss other analyses of normative conflicts and show how these may be seen as instances of the primitive patterns. In section 2.3 we identify additional cases of normative conflict which are not discussed already in the existing literature.

2.1 Primitive Patterns of Normative Conflicts

In what follows we use *Obligation*, *Permission*, *Prohibition* and *Power* as predicates that express normative relations between agents. We do not employ the axiomatization of any particular system of Deontic logic; specifically, we do not employ the axiomatization of Standard Deontic Logic (SDL), in which these notions are modeled as operators that are inter-defined. This is because in Standard Deontic Logic (and any system where the D[2] scheme is valid) it is not possible for an agent to bear conflicting

[2] $\neg O\bot$ where O denotes obligation

obligations because of the D scheme. Yet, in most realistic situations, indeed in our everyday life, agents do find themselves in normative conflict. Moreover, if we were to employ SDL, permission, obligation and prohibition would be interdefined, and so all of the patterns we present in this section (section 2.1) would be reduced to three of all six patterns (we explain this in detail in the appendix); thus the representation would be less distinguishing.

A. Conflict between a normative notion (*NN*) and its negation. The general pattern is:

$$NN(agent1, role1, action, agent2, role2)$$
$$\neg NN(agent1, role1, action, agent2, role2)$$

This is the common syntactical conflict that arises when an agent has contradictory knowledge. All other approaches, without any exception, refer to this type of conflict. In policy-based approaches, when the normative notion is *Obligation* it is called *positive-negative conflict of modalities* [21]. This type of conflict never actually arises in our representation, where norms are represented as defaults, because the derivation of extensions preserves consistency. It may, however, be identified as a potential conflict, when multiple extensions are computed.

B. Conflict between the prohibition to perform an action and the simultaneous permission or obligation to perform the same action. Once again, all previous research approaches refer to this type of conflict. In [21] and [18] these conflicts are called *conflicts between authority policies* (sub-pattern B1: Prohibition vs Permission) and *conflict between authority and imperatival policies* (sub-pattern B2: Prohibition vs Obligation) respectively.

Consider, for instance, the following default theory (W, D) where:

$W=\{Order(Agent1, RA, Agent3, WA)\}$

and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Prohibition(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : Permission(Agent3, WA, Delivery, Agent1, RA)}{Permission(Agent3, WA, Delivery, Agent1, RA)}$$ }

The first default denotes that if an order from *Agent1* (acting as retailer) towards *Agent3* (acting as wholesaler) holds, and it is consistent to assume that *Agent1* is related to a well known debtor then we may infer that *Agent3* is prohibited to perform delivery. Similarly, the second default expresses that if an order from *Agent1* towards *Agent3* holds, and it is consistent to assume that *Agent3* is permitted to perform delivery, then we may infer that *Agent3* is permitted to perform delivery towards *Agent1*. *Agent3* may find itself in a conflicting state after applying the two defaults sequentially. We denote this type of conflict as B1. Note that special terms, such as *WellKnownDebtor(agent)*, *BecomeRegularClient(agent)* or *IsRegularClient(agent)* among others, are used only for the purposes of illustration and are not binding to the characterization of domain-independent conflict patterns.

In the same spirit, let us replace the second default shown above with the following:

$$\frac{Order(Agent1, RA, Agent3, WA) : Obligation(Agent3, WA, Delivery, Agent1, RA)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

Once again *Agent3* is in conflict. We denote this conflict between Prohibition and Obligation as B2.

C. Conflict between an obligation to perform *action* and the simultaneous obligation or permission to perform ¬*action*. Here ¬*action* denotes a negative action, and the issue of representing negative actions has concerned researchers (e.g. [24] regards them as actions that do not lead to the successful fulfillment of a norm). We have not developed special semantics for the representation of negative actions; we merely regard such expressions as denoting either performance of some action other than the negative one, or as idleness (non performance of any action). This case arises, also, in Lee [17] and Abrahams [1] who use the term *Waive*.
For example consider the following DfT where:
W={*Order*(*Agent1*, *RA*, *Agent3*, *WA*)}
and *D*={

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Obligation(Agent3, WA, \neg Delivery, Agent1, RA)}$$ }

D. Conflict between the power to perform an action and the simultaneous prohibition to perform the same action. This type of conflict is also noted in [1].
For instance consider the following DfT:
W={*Order*(*Agent1*, *RA*, *Agent3*, *WA*)}
and *D*={

$$\frac{Order(Agent1, RA, Agent3, WA) : Power(Agent3, WA, Delivery, Agent1, RA)}{Power(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Prohibition(Agent3, WA, Delivery, Agent1, RA)}$$ }

One may argue that in this case there is no conflict and, consequently, that there is no need for conflict resolution. Indeed, legal power to perform an action goes hand-in-hand with permission to exercise it, according to formal definitions of institutional power ([19, 14]). Hence, there is a conflict here, albeit some may perceive it as a conflict between permission and prohibition to exercise a certain power.

E. Conflict between two obligatory distinct actions, when it is impossible to do both at the same time. This corresponds to Horty's moral dilemma [12].
For instance consider the following DfT where:
W={*Order*(*Agent1*, *RA*, *Agent3*, *WA*), *Order*(*Agent2*, *RA*, *Agent3*, *WA*),
no simultaneous performance of actions is possible}
and *D*={

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery1, Agent1, RA)}$$

$$\frac{Order(Agent2, RA, Agent3, WA, T1) : IsRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery2, Agent2, RA)}$$ }

Agent3 bears two obligations that cannot be simultaneously satisfied.

F. Conflict between an obligation and the negation of the agent's permission or power to perform it. The negation of an agent's permission/power to perform an action may be explicitly derived from the agent's knowledge base (sub-pattern F1) or it may be derived from a possibly incomplete knowledge base, through the absence of explicit information (sub-pattern F2).
For instance consider the following default theory where:
$W=\{Order(Agent1, RA, Agent3, WA)\}$
and $D=\{$

$$\frac{Order(Agent1, RA, Agent3, WA) : BecomeRegularClient(Agent1)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{\neg Permission(Agent3, WA, Delivery, Agent1, RA)}\ \}$$

Now consider a DfT that contains the first of the defaults above and in place of the second, the following:

$$\frac{Order(Agent1, RA, Agent3, WA) : \neg Permission(Agent3, WA, Delivery, Agent1, RA)}{\neg Permission(Agent3, WA, Delivery, Agent1, RA)}$$

If the agent's knowledge base does not contain an explicit permission, then the justification of this default will be satisfied, and hence its conclusion will be drawn.

2.2 Other Analyses of Normative Conflicts

In this section we review some of the main ideas that other researchers have proposed in their analyses of normative conflict and discuss how these may be regarded as instantiations of the primitive patterns presented in the previous section. Although all the patterns discussed in this section may be regarded as special cases of the primitive patterns we introduced, they merit a separate discussion because they contain additional information that may be useful for efficient conflict resolution. We do not show the entire DfT, where it is clear that the conflicting norms arise as a result of distinct default rules.

2.2.1 Policy-based Conflicts

Intra-policy conflicts. Dunlop *et al.* [7] refer to an *internal policy conflict*, when contradictory policies are assigned to a single role. A policy in their approach corresponds to what we call a single norm.

Consider, for example, the two distinct obligations of *Agent3* (a wholesaler) to perform delivery towards two distinct retailers (*Agent1* and *Agent2*).

Obligation(Agent3, WA, Delivery1, Agent1, RA)
Obligation(Agent3, WA, Delivery2, Agent2, RA)

The conflict arises from the fact that contradictory policies are assigned to *Agent3* when acting as wholesaler. Apparently, this specific case can be mapped onto pattern

E. In the same manner, other examples of this kind may be seen as instances of other primitive patterns.

Inter-policy conflicts. Dunlop *et al.* [7] refer to an *external policy conflict*, when an agent simultaneously assumes different roles that contradict "in co-existence".

Consider, for example, that when *Agent3* acts as a wholesaler it is obliged to perform delivery towards *Agent1* while when it acts as a mediator it is prohibited to perform the same action.

Obligation(Agent3, WA, Delivery, Agent1, RA)
Prohibition(Agent3, MA, Delivery, Agent1, RA)

This specific example can be mapped onto pattern B2.

2.2.2 Role-based Conflicts

Intra-role conflicts. Cholvy *et al.* [4], consider conflicts only among different roles. In their approach a role is defined through a set of consistent norms. We believe that for a variety of applications it is not realistic to insist on consistent role definitions, and thus we accept intra-role conflicts. Typical examples of this kind of conflict are *authority conflicts* [21] and conflicts that are related with the notion of power.

Consider the case where *Agent3* who acts as a wholesaler is both permitted and prohibited to perform delivery towards the retailer *Agent1*. This inconsistency may arise depending on the assumptions that are made, such as the ones presented earlier on the relation of the retailer with a well known debtor.

Permission(Agent3, WA, Delivery, Agent1, RA)
Prohibition(Agent3, WA, Delivery, Agent1, RA)

Apparently, this case can be mapped onto pattern B1.

Inter-role conflicts. Cholvy *et al.* [4] and Dunlop *et al.* [7] identify an inter-role conflict when contradictory norms arise as a result of multiple roles being assigned to an agent.

For example, when *Agent3* acts as a carrier it is obligatory to perform delivery. If, at the same time, the same agent assumes the role of wholesaler, then such delivery is not obligatory.

Obligation(Agent3, CA, Delivery, Agent1, RA)
¬Obligation(Agent3, WA, Delivery, Agent1, RA)

This case can be mapped onto pattern A.

Obviously intra-policy and intra-role conflict patterns, as well as inter-policy and inter-role conflict patterns are conceptually related. The respective authors use the terms "policy" and "role" differently, and the only reason for discussing them separately is to facilitate comparison.

2.2.3 Conflicts related to Interest/Duty

Conflicts of interest. Moffett *et al.* in [21] define *conflicts of interest* as the situation where "the same subject can perform management tasks on two different sets of targets". This type of conflict can be seen as an instance of the inter-role conflict or the inter-policy conflict or the primitive E (conflict between two obligations).

Conflicts of Duty. Moffett *et al.* in [21] and later Lupu *et al.* in [18] define *conflicts of duties* and *application specific conflicts* respectively. They refer to situations where the same agent should not be allowed to perform two distinct actions (e.g. the same agent should not be allowed both to enter a payment and to sign the payment

cheque). Such conflicts may be seen as instances of the inter-role conflict or the inter-policy conflict or the primitive E (conflict between two obligations).

2.2.4 Exceptions

This type of conflict arises generally in norm-governed systems. As Sartor [25] notes such conflicts emerge when "exceptions to norms state that particular norms, unambiguously identified, do not apply in a given situation".

Consider the following DfT where the retailer *Agent1* who holds a discount card orders goods from the wholesaler *Agent3*. Based on the first default the retailer gets a 10% discount due to the discount card. On the other hand, based on the second default the retailer should get a 20% discount because it places an order during the sales period. The described conflict is of type E:

Obligation(Agent3, WA, Discount10%, Agent1, RA)
Obligation(Agent3, WA, Discount20%, Agent1, RA)

2.2.5 Temporal Normative Conflicts

Dunlop *et al.* [7] present a temporal logic based approach for the detection of normative conflicts. In this section we present briefly a modification of our representation of normative relations in DfL, which takes into account the *external* time of a norm (i.e. the time at which it comes into force) and the *internal* time of a norm (i.e. the time stipulated for its satisfaction, its deadline) (cf. [20]). A formula of the form:

NN(agent1, role1, action, time2, agent2, role2, time1)

denotes that at time point *time1* *agent1* (acting as *role1*) is in legal relation *NN* towards *agent2* (acting as *role2*) to perform *action* by *time2*.

Now, we may discuss normative conflicts of the types described by the primitive patterns B-F, in a temporal setting. For the purposes of illustration consider the primitive pattern E, in which the following norms are in conflict:

Obligation(Agent3, WA, Delivery, IT1, Agent1, RA, ET1)
Obligation(Agent3, WA, Delivery, IT2, Agent2, RA, ET2)

where $IT1$, and $ET1$ are the internal/external time points for the first norm, and $IT2$, $ET2$ are the internal/external time points for the second norm. Temporally well formed norms are those whose internal time is subsequent to their external time, so each normative proposition corresponds to an interval; the intervals for the example we use here are $\Delta T1=[ET1, IT1]$ and $\Delta T2=[ET2, IT2]$.

A conflict arises in the following situations:

(i) $ET1 = ET2$ and $IT1 = IT2$: when $\Delta T1$ coincides with $\Delta T2$.
(ii) $ET1 \leq ET2 < IT2 \leq IT1$: when $\Delta T1$ fully overlaps $\Delta T2$.
(iii) $ET1 < ET2 < IT1 < IT2$: when $\Delta T1$ partially overlaps $\Delta T2$.
(iv) $IT1 = ET2$: when $\Delta T1$ meets $\Delta T2$. This conflict holds only at time point $IT1 = ET2$.

Note that for completeness, one should also consider the symmetrical cases.

2.3 Additional Patterns

Here are some additional cases of normative conflict that we have identified, which are not discussed already in the existing literature. We mention them separately be-

cause, although they may be reduced to the primitive patterns, there is additional information that may be exploited to facilitate conflict resolution.

Type of action-based conflicts. A common feature of e-contracts is the so called Contrary-to-Duty structures [22]. An agent's contractual obligations may be divided in two types. *Prima facie* obligations that concern the performance of actions that are in principle stipulated by the agreement and secondary obligations that concern the performance of reparatory actions; the latter apply only when violations of *prima facie* obligations happen.

An agent may, thus, bear two distinct obligations (for instance of the kind described by E), where one is primary and the other is secondary. This qualification may be helpful in conflict resolution. The general pattern is:

Obligation(agent1, role1, action, agent2, role2)
Obligation(agent1, role1, reparatoryaction, agent3, role3)

Agreement-based conflicts. An agent may find itself in a conflicting state because it is engaged in multiple contracts. For instance a wholesaler may be obliged to perform two distinct deliveries to two distinct retailers as dictated by two distinct agreements. This situation may be regarded as the generalization of pattern E discussed earlier, because in this case the important information is the distinction between the contracts. The additional information that the two norms stem from two agreements, may be exploited for the purposes of conflict resolution. The general pattern is:

Obligation(contract1, agent1, role1, action1, agent2, role2)
Obligation(contract2, agent1, role1, action2, agent3, role3)

where normative propositions of the form *NN(contract, agent1, role1, action, agent2, role2)* express that according to *contract*, *agent1* that acts as *role1* is in legal relation *NN* towards *agent2* that acts as *role2* to perform *action*.

Note that this conflict pattern is different form the one presented in [11]. The key notion here is the different contracts an agent has to comply with. Different contracts may be established towards different agents or even towards the same agent.

Conflicts between assumptions and knowledge. A conflict may arise not only as a result of an agent's explicit knowledge but also between its knowledge and its current assumptions or even between distinct assumptions.

For example, according to the following DfT the prohibition that derives from the second default contradicts not only with obligation that derives from the first default, but also with the assumption of the first default (*Permission*):

W={*Order*(*Agent*1, *RA*, *Agent*3, *WA*)}

and *D*={

$$\frac{Order(Agent1, RA, Agent3, WA) : Permission(Agent3, WA, Delivery, Agent1, RA)}{Obligation(Agent3, WA, Delivery, Agent1, RA)}$$

$$\frac{Order(Agent1, RA, Agent3, WA) : WellKnownDebtor(Agent1)}{Prohibition(Agent3, WA, Delivery, Agent1, RA)}$$ }

3 Conclusions and Future Work

We presented a set of normative conflict patterns that may be encountered in e-contracts, and discussed how other analyses of normative conflicts found in the litera-

ture of distributed systems, legal reasoning and multiagent interaction may be seen as instances of these patterns. We also identified some conflicts that have not been identified yet in other proposals.

In [10] we proposed the representation of e-contracts as default theories that can be constructed dynamically from event calculus representations because this affords us the ability to perform temporal reasoning, defeasible reasoning and conflict resolution. As regards the latter in particular, we should note that default logic coupled with priorities can be very expressive. In our previous work [9], we showed how the ascription of priorities to default rules can be done dynamically and consequently how the various strategies for conflict resolution that have been proposed in the literature can be accommodated. Our current work focuses on developing a computational tool based on Reiter's DfL and its major variations, that supports temporal defeasible reasoning as well as conflict detection and resolution as presented in this paper and in [9].

Appendix

We mentioned in section 2.1 that if one accepts Standard Deontic Logic then some of the conflict patterns that we presented are mapped onto others. Here we explain this further, and discuss why we find the adoption of such an axiomatization undesirable.

According to SDL the following inter-definability relations hold among operators for Obligation (O), Permission (P) and Prohibition (F):

It is obligatory that a: Oa
It is permitted that a: $Pa \equiv \neg O \neg a$
It is prohibited that a: $Fa \equiv \neg Pa \equiv O \neg a$

In the following table we show the conflict patterns that arise, if one adopts an SDL axiomatization:

Table 1. Conflict patterns with and without SDL axiomatization

Pattern Type	Without SDL	With SDL
A	(Oa vs $\neg Oa$) or (NNa vs $\neg NNa$)	(Oa vs $\neg Oa$) or (NNa vs $\neg NNa$)
B1	(Fa vs Pa)	(Ob vs $\neg Ob$) if $b \equiv \neg a$ is assumed
B2	(Fa vs Oa)	(Oa vs $O\neg a$) or (Oa vs Ob) if $b \equiv \neg a$ is assumed
C	(Oa vs $O\neg a$)	(Oa vs Ob) if $b \equiv \neg a$ is assumed
D	(PWa vs Fa)[3]	-
E	(Oa vs Ob)	(Oa vs Ob)
F	(Oa vs $\neg Pa$)	(Oa vs $O\neg a$) or (Oa vs Ob) if $b \equiv \neg a$ is assumed

[3] PWa denotes power to perform the action a

As can be seen from this table, under SDL inter-definability of operators, we may consider that only three primitive conflict patterns arise, i.e. type A, type D and type E.

We noted in section 2 that in the representation of norms as default theories the conflict pattern A never actually arises in an extension, because the derivation of extensions preserves consistency. But, this does not hold for pattern B1, which is possible even as an inter-extension conflict. If we accept the inter-definability of Deontic operators, pattern B1 essentially collapses and becomes pattern A. Hence, B1 will never actually arise in an extension.

Adopting an SDL axiomatization of Deontic operators seems, thus, to lead to some sort of a priori pruning of potential normative conflicts. We find this undesirable, because it seems unrealistic to assume that fewer normative conflicts may arise for agents in a virtual environment, than in real world situations. We prefer, therefore, to maintain a more discriminating representation, which can express more conflict patterns, so that these may be detected and eventually resolved appropriately.

References

1. Alan S. Abrahams and Jean M. Bacon. The life and times of identified, situated, and conflicting norms. In *Deontic Logic in Computer Science, 6th International Workshop on Deontic Logic in Computer Science*, pages 3–20, London, England, May 2002.
2. Grigoris Antoniou. A tutorial on default logics. *ACM Computer Surveys*, 31(4):337–359, 1999.
3. Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert der van Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, Montreal, Canada, 2001. ACM Press.
4. Laurence Cholvy and Frederic Cuppens. Solving normative conflicts by merging roles. In *ICAIL '95: Proceedings of the 5th international conference on Artificial intelligence and law*, pages 201–209. ACM Press, 1995.
5. Laurence Cholvy and Frederic Cuppens. Reasoning about norms provided by conflicting regulations. In Henry Prakken and Paul McNamara, editors, *Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science*, pages 247–264. IOS Press, Amsterdam, 1998.
6. Aspassia Daskalopulu. Modeling legal contracts as processes. In *Legal Information Systems Applications, 11th International Workshop on Database and Expert Systems Applications (DEXA'00)*, pages 1074–1079. IEEE Computer Society, 2000.
7. Nicole Dunlop, Jadwiga Indulska, and Kerry Raymond. Dynamic conflict detection in policy-based management systems. In *Proceedings of the Sixth International Enterprise Distributed Oblect Computing Conference*, pages 15–26, Washington, DC, USA, 2002. IEEE Computer Society.
8. Nicole Dunlop, Jadwiga Indulska, and Kerry Raymond. Methods for conflict resolution in policy-based management systems. In *Proceedings of the 7th International Conference on Enterprise Distributed Object Computing*, pages 98–111, Washington, DC, USA, 2003. IEEE Computer Society.
9. Georgios K. Giannikis and Aspassia Daskalopulu. Defeasible reasoning with e-contracts. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 690–694, Hong Kong, China, 2006. IEEE Computer Society.

10. Georgios K. Giannikis and Aspassia Daskalopulu. The representation of e-Contracts as default theories. In H.G. Okuno and M. Ali, editors, *Proceedings of 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2007*, LNAI 4570, pages 963–973, Kyoto, Japan, 2007. Springer-Verlag Berlin Heidelberg.
11. H. Herrestad and C. Krogh. Deontic logic relativised to bearers and counterparties. In *Anniversary Anthology in Computers and Law, Ed. J. Bing. and O. Torrund*, pages 453 – 522, 1995.
12. John F. Horty. Moral dilemmas and nonmonotonic logic. *Journal of Philosophical Logic*, 23(1):35–65, 1994.
13. John F. Horty. Reasoning with moral conflicts. *Nous*, 37(4):557–605, 2003.
14. Andrew J.I. Jones and Marek J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):427–443, 1996.
15. Martin J. Kollingbaum and Timothy J. Norman. Strategies for resolving norm conflict in practical reasoning. In *ECAI Workshop CEAS*, 2004.
16. Robert A. Kowalski. A logic-based model for conflict resolution, 2003. http://www.doc.ic.ac.uk/rak/.
17. Ronald M. Lee. Bureaucracies as deontic systems. *ACM Trans. Inf. Syst.*, 6(2):87–108, 1988.
18. Emil Lupu and Morris Sloman. Conflicts in policy-based distributed systems management. *IEEE Trans. Software Eng.*, 25(6):852–869, 1999.
19. David Makinson. On the formal representation of rights relations. *Journal of Philosophical Logic*, 15(4):403–425, 1986.
20. Rafaél Hernández Marín and Giovanni Sartor. Time and norms: a formalisation in the event-calculus. In *7th International Conference on Artificial Intelligence and Law*, pages 90–99, New York, NY, USA, 1999. ACM Press.
21. Jonathan D. Moffett and Morris S. Sloman. Policy conflict analysis in distributed system management. *Journal of Organizational Computing*, 1993.
22. Henry Prakken and Marek J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.
23. Raymond Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.
24. Lamber M. M. Royakkers. *Extending Deontic Logic for the Formalisation of Legal Rules*. Kluwer Academic Publishers, 1998.
25. Giovanni Sartor. Normative conflicts in legal reasoning. *Artificial Intelligence and Law*, 1(2-3):209–235, 1992.

Machine Learning

Tri-Testing: A Novel Semi-Supervised Learning Method based on Ensemble Learning and Active Learning

Shota Kubo[1], Masahiro Terabe[1], Kazuo Hashimoto[1], and Ryuta Ono[2]

[1] G.S.I.S., Tohoku University,
6-6-11-304 Aramaki-Aza-Aoba, Aoba-Ku, Sendai, 980-8579, Japan
{kubo, terabe, kh}@aiet.ecei.tohoku.ac.jp
[2] NEC Corporation
r-ono@bk.jp.nec.com

Abstract. In many practical problems such as web page classification, it is highly expensive to obtain labeled training data set. Semi-supervised learning is a method to form a training data set by labeling abundant unlabeled data with classifiers learned from a small number of labeled data. It has a remarkable advantage to reduce the cost of labeling. We propose Tri-Testing, a novel semi-supervised learning method. We improved Tri-Training, a semi-supervised learning method based on ensemble learning, by introducing active learning. Tri-testing is defined as a combination of an ensemble learning process with three different classifiers, and an active learning process which selectively choose unlabeled data set to be labeled. The experiment shows that the proposed method derives more accurate classifier than existing other method, especially in the case where only small labeled data is available as training data set.

1 Introduction

In supervised learning scheme, classifiers are trained by a set of labeled data, and its accuracy is improved as the size of training data set increases. However, providing a sufficient size of training data set is sometimes difficult because it requires human expertise in the field under consideration for accurate labeling. There are cases where a large number of unlabeled instances are easy to obtain but labeled instances are hard. Web page classification is such a case. Semi-supervised learning increases the number of labeled data by labeling the unlabeled data with classifiers by themselves, and shows a better performance than supervised learning using only the original set of training data.

The concept of semi-supervised learning started to be studied in the mid '60s[4] as a probabilistic approach, such as EM algorithm, generative model, discriminative model, and so forth. These approaches are all parametric methods, and require sufficient supply of training data.

Blum et al. have proposed a PAC learning based algorithm, Co-Training[2]. Since PAC learning is not sensitive to the distribution of cases, Co-Training

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 175–184

Received 17/07/07
Accepted 31/08/07
Final version 23/09/07

works well with a small number of labeled data, and is noise tolerant. But Co-Training works under the strong assumption that cases are to be divided into two sufficient and redundant *view*. Which means attribute set can split into two independent subsets. This constraint sometimes prevent from Co-Training being applied to real problem.

Several methods are derived from Co-Training. Co-Testing[5] introduces an active-learning technique into Co-Training and Co-EM uses EM with Co-Training. Zhou et al.[8] proposes Tri-Training, an ensemble learning with three classifiers, each of which is trained by bootstrap sampling of labeled instance set. Tri-Training avoids constraint on Co-Training.

In this paper, we propose Tri-Testing, a novel semi-supervised learning method inspired by Tri-Training. Tri-Testing is an active learning process which selects unlabeled data to improve the accuracy of classification using three classifiers.

The following section is composed as follows. Section 2 discusses the conventional methods and Section 3 presents the proposed method. Section 4 and 5 shows our experiment that compare with other methods. Finally, Section 6 is the conclusion and future work.

2 Semi-supervised Learning

Semi-supervised learning trains classifiers using a set labeled data, as supervised learning does. Classifiers predict a label of each unlabeled data, selectively add predicted data, which is expected to improve the performance the most, to the original training data set, then updates classifiers with an updated training data set. The process iterates until the performance saturates.

Algorithms are different in according to the followings;

- How to generate classifier
- How to select unlabeled instance to label
- How to make final prediction

2.1 Co-Training

Co-Training[2] splits an attribute set of labeled data into two independent sets A_1 and A_2 which authors called as *"view(s)"*, and divide the labeled data into S_1 and S_2 according to the views. The original label y is given to splitted instances. Then, two classifiers h_1 and h_2 from each labeled data sets S_1 and S_2 are induced. Each classifier h_1 and h_2 predicts a label of any instance x in unlabeled dataset U, and allows h_i to label instances selectively where high confidence is obtained. Finally, classifiers are updated with both S_1 and S_2 and self-labeled instances. The whole process iterates k (k is an optional parameter) times.

Co-Training is theoretically proven that unlabeled instance supports classify ability. However, there are problem about applicability in real data because it is sometimes difficult to define view appropriately even though the view definition is a key performance factor of Co-Training. Additionally, some of the data set does not include sufficient number of attributes to construct view.

2.2 Co-Testing

Co-Testing[5] introduces an active learning approach to Co-Training based on the idea that active learning works better with multiple views than a single. Co-Testing selects instances with higher confidence from a set of instances whose predicted labels are not consistently given by two classifiers, and asks human experts to label and teach them.

Co-Testing improves effectively by collaborate with human expert. However Co-Testing also requires many attributes, as Co-Training does.

2.3 Tri-Training

Co-Training requires many attributes to define views appropriately. To make free to this constraints, Tri-Training[8] employs an ensemble learning approach like Bagging[3] with three classifier.

First, Tri-Training prepares three different and same size data sets S_1, S_2, and S_3 by bootstrap resampling the labeled data set L, and generates classifiers h_1,h_2, and h_3 respectively. In labeling phase, each classifier predicts a label for unlabeled instance u if two classifiers predict the same label. For instance, h_1 predicts a label y for an unlabeled instance u where $h_2(u) = h_3(u) = y$. Then classifiers are retrained with using both self-labeled and originally labeled instances.

Tri-Training shows a better classification performance than Co-Training if dataset has insufficient number of attributes. Tri-Training creates a labeled instance and add to the labeled instance set if two other classifiers, h_2, h_3, agree, regardless of the prediction by the classifier h_1. It is expected that the probability for the classifier h_1 to agree with h_2 and h_3 is not negligibly small, the performance improvement will saturate.

3 Tri-Testing

We propose Tri-Testing, a novel semi-supervised learning method by introducing an active learning approach to Tri-Training. As Tri-Training does, Tri-Testing generates three classifiers h_1,h_2, and h_3 from S_1, S_2, and S_3 which are sampled from labeled data set L by bootstrap resampling.

Tri-Testing asks human expert to label some of the instances that prediction disagrees among classifiers, and retrain classifiers with updated labeled instance set. Tri-Testing is designed to reduce this labeling cost of human expert by selecting as few unlabeled instances as possible to improve the overall accuracy in classification.

3.1 Concept of Tri-Testing

For instance, think about learning from dataset which has two class, "YES" or "NO". After initial learning, the number of combination of prediction from three classifier h_1,h_2, and h_3 is 2^3, as shown by table 1.

Table 1. Combination of label prediction by three classifiers.

Region	h_1	h_2	h_3
R_1	YES	YES	YES
R_2	YES	YES	NO
R_3	YES	NO	YES
R_4	YES	NO	NO
R_5	NO	YES	YES
R_6	NO	YES	NO
R_7	NO	NO	YES
R_8	NO	NO	NO

By predictions of three classifier h_1,h_2, and h_3, instances are classified into 8 region, R_1 to R_8. R_1 and R_8 regard as classify instance correctly because all classifiers predict the same label. The classifiers have already learned the instance label correctly so that labeling instance in R_1 or R_8 and retraining classifier may not boot classify accuracy significantly. The other hand, label prediction for instance in R_2 to R_7 is not agreed among all three classifiers. In this case, we can assume that some of the classifiers have not learned the instance label efficiently. Therefore, if human teach the label to such instance, at least one of classifier's performance probably refine.

3.2 Tri-Testing Algorithm

In this section, we explain how Tri-Testing works referring to the pseudo code of Tri-Testing algorithm shown in table 2.

Let x be the instance, y the label by human expert, l the true label, L the labeled data set, U the unlabeled data set, *Learn* the learning algorithm, S_i the bootstrap resampled L, h_i the classifier induced from S_i, $ContentionPoints[i]$ the list of candidate for label, L_i the labeled instance, and *saveU* the buffer for contain unlabeled instance.

At first, Tri-Testing prepares an initial training data set with different class distribution S_1, S_2, and S_3 by bootstrap resampling from initial labeled data L. Three classifiers h_1, h_2, and h_3 are generated from data set S_1, S_2, and S_3. The classifiers h_1, h_2, and h_3 are slightly different with each other because the training data set S_1, S_2, and S_3 has different instance composition.

The next step is a selection of instance for human labeling, which is the most important phase in Tri-Testing. Each classifier h_i predicts the label of every instance x in unlabeled data set U. If only one classifier h_i predicts the label of instance x differently among three classifiers, the instance x is added to the $ContentionPoints[i]$. The instances in $ContentionPoints[i]$ are candidates of asking the label to human expert.

Classifier h_i can improve its prediction accuracy when a correct label is given to the instance for which h_i predicts wrongly with high confidence. Tri-Testing select an instance has most confidence by h_i from $ContentionPoints[i]$ list for

Table 2. Pseudo code describing the Tri-Testing Algorithm

Input: L, U, $Learn$
for $i \in \{1..3\}$ **do**
  $S_i \leftarrow BootstrapSample(L)$ //bootstrap resample
  $h_i \leftarrow Learn(S_i)$ //generate h_i
end of for
for k iterations //k is optional number
  for $i \in \{1..3\}$ **do**
    $ContentionPoints[i] \leftarrow \phi$
  end of for
  for $i \in \{1..3\}$ **do**
    for every $x \in U$ **do** $(j, k \neq i)$
      if $h_i(x) \neq h_j(x)$ and $h_j(x) = h_k(x)$
      then $ContentionPoints[i] \leftarrow x \ \ ContentionPoints[i]$
    end of for
    $x \leftarrow SelectQuery(ContentionPoints[i])$
    $y \leftarrow Label(x)$ //hand-ladel by human
    $delete(x, ContentionPoints[i])$
    if $y \neq h_i(x)$
    then $L_i \leftarrow L_i\{(x, y)\}$
    else then $L_j \leftarrow L_j\{(x, y)\}$
      $L_k \leftarrow L_k\{(x, y)\}$
  end of for
  for $i \in \{1..3\}$ **do**
    $saveU \leftarrow saveUContentionPoints[i]$
    $h_i \leftarrow Learn(LL_i)$
  end of for
  $U \leftarrow saveU$
end of for k iterations
Output: $h(x) \leftarrow \arg\max_{l \in label} \sum_{i:h_i(x)=y} 1$ //majority vote

retraining. An instance which has high confidence may locate in far from class bound.

In the case shown in table 1, this method has the following assumptions;

- h_1 makes wrong predictions for instances in R_4 and R_5,
- h_2 makes wrong predictions for instances in R_3 and R_6,
- h_3 makes wrong predictions for instances in R_2 and R_7.

An instance x in $ConjunctionPoint[i]$ with highest confidence of h_i is picked out. The instance x is labeled by human expert. If the label of instance x predicted by $h_i(x)$ and human expert y is different, the instance x is added to the training set L_i. If the $h_i(x)$ and y is the same label, it indicates that h_j and h_k predicts wrong class. In this case, the instance x is added to L_j, L_k for retraining.

Finally, three classifiers retrained with labeled data set L_i and original instance set L. Tri-Testing iterates this process specified times, and generates final

classifiers. The iteration also stops in the case all *ContentionPoints* are empty, because it assume there is no instance efficient for improving the classifier.

4 Experiment 1: Unlabeled Instance Rate and Prediction Accuracy

4.1 Exprimental Setting

To confirm the efficiency of Tri-Testing, we conducted the following experiments. We used dataset from UCI Machine Learning Repository[1]. All data sets are binary class, and are used in the experiment of [8]. Table 4 indicates specifications of each dataset. "Pos." and "Neg." shows proportions of positive and negative instances in the data set respectively.

Table 3. The specifications of data sets for experiment.

Dataset	# of inst.	# of attr.	# of cls.	Pos./Neg. (%)	Cont. Attr. (%)	Missing Value
AUSTRALIAN	690	14	2	55.5/44.5	42.8	no
BUPA	345	6	2	42.0/58.0	100.0	no
COLIC	368	22	2	63.0/37.0	31.8	yes
DIABETES	768	8	2	65.1/34.9	100.0	no
GERMAN	1000	20	2	70.0/30.0	35.0	no
IONOSPHERE	351	34	2	35.9/64.1	100.0	no
KR-VS-KP	3196	36	2	52.2/47.8	0.0	no
SICK	3772	39	2	6.1/93.9	19.4	yes
TIC-TAC-TOE	958	9	2	65.3/34.7	0.0	no
VOTE	435	16	2	61.4/38.6	0.0	yes

We compare the prediction accuracy of following three learners.

- **NB (Naive Bayes):** Single classifier induced by Naive Bayes.
- **Tri-Training:** Ensemble of three classifiers. Each classifiers induced by Naive Bayes.
- **Tri-Testing:** Ensemble of three classifiers. Each classifiers induced by Naive Bayes. Maximum iteration times $k = 5$.

In the first experiment, we investigate the effect of unlabeled instance rate on prediction accuracy. In order to take into account the influence of the sampling in Tri-Training and Tri-Testing ten fold cross validation was repeated 10 times and the error rate was evaluated by taking the average over the traials.

4.2 Results

Table 4 indicates error rate of each learners. In all of the data set, the error rate is not so different among all learners when the unlabeled instance rate is under the 90%. However, in most of the cases, Tri-Testing shows better performance than the other learners when the unlabeled instance rate is over the 90%.

We also run a paired t-test between Tri-Training and Tri-Testing. The prediction accuracy of Tri-Testing performs better in the significant level over the 95% is bolded. In AUSTRALIAN, COLLIC, IONOSPHERE, and KR-VS-KP, Tri-Testing performs better than other algorithms with statistically significant level over 95%.when the unlabeled instance rate is over 95%. Additionally, in DIABETES, GERMAN, TIC-TAC-TOE, and VOTE, Tri-Testing outperforms at confidence level 95% when unlabeled instance rate is over 99%.

4.3 Discussion

In most of the cases, Tri-Testing performs better than Tri-Training when the unlabeled instance rate high. This results shows that active learning works well to improve the prediction accuracy of each classifiers and ensemble classifier.

In the case of BUPA, IONOSPHERE, Tri-Testing and Tri-Training perform worse than Naive Bayes when the unbalanced instance rate is 99%. This is because that the number of training data for weak learners in ensemble. The weak learners learn from the training data which includes only 3 or 4 instances. This effect is investigated in our former research[7].

On the other hands, in the case of SICK, Tri-Testing can not improve the prediction accuracy, and Tri-Training also can not. SICK's class distribution is extremely biased. In this data set, the ratio of positive class is only 10%. We assume that this biased class distribution makes it hard to learn for ensemble learner such as Tri-Training and Tri-Testing.

5 Experiment 2: Class Distribution and Prediction Accuracy

5.1 Experimental Setting

As confirmed in experiment 1, Tri-Testing does not perform well when the class distribution is biased as SICK data set. If the class distribution bias is canceled, the performance of Tri-Testing would be improved. Gao et al.[9] demonstrated the effect of class distribution bias canceling by sampling. In this experiment, we investigate the effect of class distribution bias to the prediction accuracy of proposed method.

We use the same data set as experiment 1. We compare two cases;

- **Case1 (Uniform Class Distribution):** The training data for weak learners in ensemble is sampled to be the class distribution uniformly.

Table 4. The classification error rate of NaiveBayes, Tri-Training and Tri-Testing.

Dataset	Method	Unlabeled Instance Rate(%)							
		20	40	60	80	90	95	97	99
AUSTRALIAN	NB	22.8	22.9	22.6	22.3	21.2	22.8	24.7	31.5
	Tri-Training	22.8	22.9	22.6	22.3	21.0	23.1	24.0	28.0
	Tri-Testing	22.6	22.7	22.5	21.7	20.7	**21.4**	**22.3**	**23.6**
BUPA	NB	44.7	44.3	43.5	46.7	46.1	45.5	46.6	48.5
	Tri-Training	45.9	45.5	44.3	47.1	46.7	46.3	46.7	42.7
	Tri-Testing	44.3	44.1	43.3	46.2	46.2	45.3	45.5	46.3
COLIC	NB	21.5	21.8	21.7	23.3	25.3	29.5	33.2	49.5
	Tri-Training	21.6	21.9	21.8	22.9	24.9	29.5	30.8	54.6
	Tri-Testing	21.4	21.6	21.3	22.2	23.0	**25.7**	**27.7**	**39.8**
DIABETES	NB	24.8	25.0	25.1	26.2	27.4	29.4	31.4	34.9
	Tri-Training	24.7	25.0	25.1	26.2	27.5	29.3	31.5	35.0
	Tri-Testing	24.8	24.9	24.8	26.0	27.0	28.3	30.2	32.4
GERMAN	NB	25.1	25.3	25.8	26.8	29.3	31.0	32.5	37.0
	Tri-Training	25.2	25.3	25.9	26.9	29.2	30.9	32.7	35.2
	Tri-Testing	25.0	25.3	25.8	26.7	28.9	30.7	32.3	**33.8**
IONOSPHERE	NB	17.9	17.8	18.1	18.1	18.8	19.0	26.9	36.3
	Tri-Training	18.0	18.1	18.8	19.0	21.1	22.3	30.5	35.4
	Tri-Testing	17.3	17.2	17.4	16.3	**16.2**	**16.3**	**21.4**	37.0
KR-VS-KP	NB	12.4	12.8	13.0	13.7	16.5	18.5	20.3	28.0
	Tri-Training	12.4	12.8	13.1	13.8	16.6	18.8	20.9	31.0
	Tri-Testing	12.4	12.7	12.9	13.6	15.9	**17.6**	**19.0**	**25.5**
SICK	NB	7.3	7.3	7.7	7.8	7.6	6.4	6.3	6.8
	Tri-Training	7.3	7.3	7.8	7.9	7.7	6.6	6.4	7.0
	Tri-Testing	7.3	7.4	8.0	8.2	8.5	7.5	7.7	7.2
TIC-TAC-TOE	NB	29.7	29.6	29.8	30.5	32.4	34.6	36.4	40.7
	Tri-Training	9.8	29.6	29.8	30.5	32.6	34.8	36.7	41.1
	Tri-Testing	29.6	29.6	29.9	30.4	32.3	33.9	35.7	**37.4**
VOTE	NB	9.7	9.8	9.9	9.9	10.0	10.4	10.2	11.9
	Tri-Training	29.7	9.7	9.8	10.0	10.2	10.6	10.6	11.7
	Tri-Testing	9.9	9.8	9.7	9.7	9.6	9.7	9.3	**8.8**

– **Case2 (Original Class Distribution):** The training data for weak learners in ensemble is sampled randomly. The class distribution of training data is not different from that of labeled data set.

We run Tri-Testing in this two cases and evaluate the classification error rate in various unlabeled instance rate.

5.2 Results

We calculate the difference of error rate between case 1 and 2. If the difference is negative, the prediction accuracy is improved by canceling the class distribution bias by sampling. The experimental result is shown in table 5.

Table 5. Effect of canceling the class distribution bias.

Dataset	Unlabeled Instance Rate(%)			
	90%	95%	97%	99%
AUSTRALIAN	-0.1	1.37	1.10	9.42
BUPA	0.41	-0.88	-0.44	8.94
COLIC	0.86	1.36	2.22	-5.06
DIABETES	0.05	0.86	0.65	2.08
GERMAN	-0.17	-0.24	0.58	-0.88
IONOSPHERE	0.18	2.26	2.32	5.29
KR-VS-KP	0.11	0.43	0.55	1.08
SICK	-0.32	-0.58	-0.56	-0.17
TIC-TAC-TOE	0.02	-0.19	0.06	1.11
VOTE	0.10	0.07	0.63	0.98

The effect of prediction accuracy improvement is different among data set. However, in case of SICK which includes biased class distribution, the prediction accuracy is improved when the bias is canceled by sampling. This result shows the possibility that the prediction accuracy is improved by canceling the class distribution bias in the case the class distribution of original data set is biased.

6 Conclusion

Some of the interesting issues to be investigated remain. One of the issues is improving the active learning process to minimize the labeling cost of human expert and to maximize the performance. Using the confidence value of each classifier to select the label request instance is one of the promising solutions.

The other issue is design of ensemble classifier which includes adequate variance. In Co-Training, independent view is pre-defined to induce two variant classifiers. In the other hands, Tri-Training and Tri-Testing introduce bootstrap

resampling to generate slightly variant three training data set and classifiers. Introducing feature selection and construction mechanism [10] to generate adequate view automatically is interesting and to be investigated.

References

1. C. Blake, E. Keogh, C. J. Merz, "UCI repository of machine learning databases", *Department of Information and Computer Science, University of California, Irvine, CA*, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. A. Blum, T. Mitchell, "Combining labeled and unlabeled data with co-training", *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp.92–100, 1998.
3. L. Breiman, "Bagging Predictors", *Machine Learning*, Vol.24, No.2, pp.123–140, 1996.
4. H. J. Scudder, "Probability of error of some adaptive pattern-recognition machines", *IEEE Transactions on Information Theory*, Vol.11, pp.363–371, 1965.
5. I. Muslea, S. Minton, C. A. Knoblock, "Selective Sampling With Naive Co-Testing: Preliminary Result", *The ECAI-2000 workshop on Machine Learning for information extraction*, 2000.
6. K. Nigam, R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training", *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pp.86–93, 2000.
7. M. Terabe, T. Washio, H. Motoda, "S^3*Bagging*: Fast Classifier Induction Method with Subsampling and Bagging. *Proceedings of Fourth International Symposium of Intelligent Data Analysis*, pp. 177-186, 2001.
8. Z. H. Zhou, M. Li, "Tri-training: exploiting unlabeled data using three classifiers", *IEEE Transactions on Knowledge and Data Engineering*, vol.17, pp.1529–1541, 2005.
9. J. Gao, W. Fan, J. Han and P. S. Yu, "A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributionsh, *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.
10. H. Liu, H. Motoda, "Feature Selection for Knowledge Discovery and Data Miningh, Kluwer, 1998.

Statistical Selection of Relevant Features to Classify Random, Scale Free and Exponential Networks[1]

Tania Turrubiates López[1,2], Claudia Gómez Santillán[1,3], Laura Cruz Reyes[2] Rogelio Ortega Izaguirre[3], Eustorgio Meza Conde[3]

[1] Instituto Tecnológico de Ciudad Madero (ITCM). 1ro. de Mayo y Sor Juana I. de la Cruz s/n CP. 89440, Tamaulipas, México.Teléfono: 01 833 3574820 Ext. 3024.
[2] Instituto Tecnológico Superior de Álamo Temapache (ITSAT). Carretera Potrero de Llano – Tuxpan Km. 6.5, Xoyotitla, Mpio. Alamo Temapache, Veracruz, México.Teléfono: 01 756 8440038
[3]Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (CICATA). Carretera Tampico-Puerto Industrial Altamira, Km. 14.5. Altamira, Tamaulipas. Teléfono: 01 833 2600124.

tania_251179@hotmail.com, cggs71@hotmail.com, lcruzreyes@prodigy.net.mx, rortegai@ipn.mx, emezac@ipn.mx.

Abstract. In this paper a statistical selection of relevant features is presented. An experiment was designed to select relevant and not redundant features or characterization functions, which allow quantitatively discriminating among different types of complex networks. As well there exist researchers given to the task of classifying some networks of the real world through characterization functions inside a type of complex network, they do not give enough evidences of detailed analysis of the functions that allow to determine if all of them are necessary to carry out an efficient discrimination or which are better functions for discriminating. Our results show that with a reduced number of characterization functions such as the shortest path length, standard deviation of the degree, and local efficiency of the network can discriminate efficiently among the types of complex networks treated here.

1 Introduction

Any natural or artificial complex system of the real world as Internet, can be modeled as a complex network, where the nodes are elements of the system and the edges interactions between elements [1][2]. The term complex network refers to a graph with a non trivial topologic structure, this has motivated the study of topological characteristics of real networks [3][4] to identify characteristics that allow the discrimination among different types of complex networks and in this way optimize the performance of processes carried out in this networks as: search of distributed resources [1][5], traffic management and design of routing algorithms.

[1] This research was supported in part by CONACYT and DGEST

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 185–192

Received 17/07/07
Accepted 31/08/07
Final version 23/09/07

Up to now the best way to identify the type of complex network has been observing the graphic of the degree distribution. This work presents a statistical selection of a set of characterization functions that allow quantitatively identified the type of complex network.

1.1 Characterization Functions of Complex Networks.

The characterization functions provide information about the topological characteristics of a complex network, when they are applied on a network, a vector of characteristic is obtained [6]. Through this vector, the behavior of the complex networks is characterized and analyzed. The characterization functions used in this work were: average degree of the network (Avg), standard deviation of the degree (Std), shortest path length (L), diameter (D), clustering coefficient (CG), global efficiency (EG) and local efficiency (EL), these characterization functions are described and detailed in [6].

One characterization function widely used to identify the type of complex network is the degree distribution which provides graphic information about the connectivity behavior of the nodes; through this function different types of networks as random networks, scale-free networks and exponential networks can be identified.

1.2 Three Types of Complex Networks.

Toward ends of the 50's and beginnings of 60's Erdös and Rényi focused in studying statistical properties in graphs where the connection among the nodes is established randomly, these graphs are called random networks and they are characterized for approaching a degree distribution binomial when the number of nodes n is small; when $n \to \infty$, the degree distribution approaches a Poisson distribution, in these graphs the nodes have approximately the same degree, nearby to the average degree of the network [7].

In the decade of the 90's diverse researchers [8][9], discovered networks of the real world as Internet exhibits a degree distribution following a power law distribution, these networks are called scale-free, because independently of the scale (number of nodes) the main characteristic of the network does not change, a reduced set of nodes have a very high degree and the rest of the nodes have a small degree [10][11].

Some natural networks exhibit an exponential distribution [12][13], where the majority of the nodes have a degree closer to the average degree and other nodes with a high degree can be observed, these networks are called exponential networks. To analyze the characteristics of these networks and to understand the phenomena carried out in them, generation models of complex networks have been created.

1.3 Generation Models of Complex Networks

The generation models of networks are an important tool to reproduce graphs sharing topological characteristics of networks of the real world making possible its study [11]. Some generation models are the Erdös-Rényi (ER) model [7] that reproduces random networks, the Barabási-Albert (BA) model [14] that reproduces scale free networks and the Liu model [15] that reproduces scale free and exponential networks.

2 Related Work

A recent application in the field of complex networks, is used information obtained through the characterization functions with the objective to discriminate among different types of complex networks, this application is related with the area of pattern recognition also known as classification. The classification of naturals and artificial structures modeled as complex networks implicate an important question: what characterization functions to select in order to discriminate among different types of complex networks [6].

In the classification area the selection of features (in this case, characterization functions) has great benefits: to improve the performance of classification procedure, and to construct simple and comprehensible classification models, these are achieved leaving aside, irrelevant and redundant characteristics that can introduce noise [16][17].

The work reported in [6], used a classification procedure to identify the type of a network with unknown nature, the results show that the type of network assigned to the networks, vary according to the characterization functions selected and an excessive number of characterization functions can compromise the quality of the classification.

An experimentation to evaluate the stability and separability of different types of networks is presented in [18], taking into account 6 topologies of networks to discriminate, 47 characterization functions are used as inputs to the classifiers. This work does not present evidences of a selection of relevant and not redundant characterization functions.

In [19], the objective is to determine from a set of models those describing more adequately networks that represent biological systems, a technique described in [20] was utilized for extracting characteristics serving as inputs to the classifier permitting to relate an instance generated by a model with a real instance. Though the results of classification obtained are good, they do not show if the characteristics extracted by the technique improve the performance of the classifier regarding the information provided by characterization functions extensively used in the field of the physics and the social sciences. In [21] networks in 3 types of topology are classified by means of a neural network; the eigenvalues of the adjacency matrix are utilized as inputs of the neural network.

3 Experimentation

Following a general scheme of procedures described in [22] an experiment was designed, to determinate, given a set of characterization functions, the functions permitting quantitatively discriminate among three types of complex networks: random networks, scale free networks and exponential networks.

In other words, 3 different populations are defined for each type of networks, from which topological characteristics are extracted, if the averages of those characteristics are significantly different and the populations do not overlap then those characteristics can help to distinguish networks of those populations. Three possible results of classifying networks can be observed:

- Case 1: There are significant differences among the 3 types of networks according to a set of characteristics, in this way through these characteristics a new network can be classified inside a type of network.
- Case 2: There exist significant differences among some of the types of networks according to a set of characteristics.
- Case 3: There does not exist significant differences among the 3 types of networks according to a set of characteristics, these two last cases lead to wrong classifications of new networks that need to be identified.

Instances of complex networks were generated to carry out the experimentation, through the models of E-R (random networks), BA (scale free networks) and Liu (exponential networks), with 200, 512 and 1024 nodes by each type of network; subsequently topologic characteristics were extracted such as: average degree (Avg), standard deviation of the degree (Std), clustering coefficient (CG), global efficiency (EG), local efficiency (EL), the shortest path length (L), diameter (D). The statistical packages MINITAB and SAS were utilized to study these characteristics.

In the related works, the networks have the same number of nodes and edges, so the average degree is equal for each type of network; in this experimentation the number of edges with the purpose of introducing variability to the experiment and carry to correct conclusions is not determined. In [18] it was observed that determining the number of edges aid to understand phenomena of interest but introduces dangers in statistical tests.

In this way, two factors that can influence in the process of characterization were identified; these factors can be controlled without affecting data normality: the type of network and the number of nodes. The factor, type of network, is composed of 3 levels, represented by the 3 types of networks being analyzed. The factor, number of nodes, is also composed of 3 levels, 200, 512 and 1024 nodes. Giving the characteristic of the problem a two-factor factorial design was chosen. The significance level α was set in 0.05.

The operation characteristics curves was used to determine the number of instances of networks n, appropriate for detecting significant differences. Through this procedure it was determined that with n = 4 instances a probability of 98% is obtained to detect differences up to 0.01 this procedure can be found in [22]. Due that this experiment was extended to the classification, n = 4 was taken as the minimum number of instances to generate; n = 30 instances was set for the experimental design.

According to the fixed effects model describing to a two-factor factorial design [22] a hypothesis were formulated to detect significant differences. A Multivariate Analysis of Variance General (GML) was carried out to obtain results permitting to reject or to accept the hypothesis, these results are discussed subsequently.

Table 1. F_0 values calculated by GLM for each variable.

	Avg	Std	CG	EG	EL	L	D
Type of network	178.47	124.39	220.18	507.82	55.21	566.56	523.72
Number of nodes	32.80	50.45	1.70	10.50	1.07	39.11	21.07
Interaction	32.70	20.87	0.56	3.57	0.18	13.90	10.01

The values showed in Table 1, were compared with the criteria for rejection associate to the hypothesis formulated, it can be observed that according to the type of network, the characterization functions satisfy the criteria for rejection, therefore, it can be concluded that all the characterization functions differ significantly according to the type of network.

For the effect of the number of nodes, and the effect of the interaction between the type of network and number of nodes, the criteria for rejection are satisfied for Avg, Std, EG, L and D, therefore, it can be concluded that these characterization functions differ significantly according to the number of nodes presented in the network, and the interaction between the type of network and the amount of nodes.

The analysis of the residuals of Avg and CG, showed abnormalities, this is because in spite of the fact that the number of edges was not determined, Avg and CG for scale free and exponential networks, were defined by initial parameters employed by the models of generation, on the other hand the plots of Std, EL, EG, L, D showed normality.

MANOVA tests shown in Table 2, for the type of network, the quantity of nodes and the interaction, are statistically significant, by which the significant differences detected in the analysis of a variable at one time are reaffirmed, being real and not false positive. Due that the values of F0 are greater for factor of type of network, it is concluded that this factor has greater influence in the values obtained by the characterization functions.

Table 2. F_0-values calculated for the MANOVA test and values of F distribution.

MANOVA Test	Type of network		Number of nodes		Interaction	
	F_0	F	F_0	F	F_0	F
Wilks'	515.91	1.23	88.62	1.23	32.97	1.17
Pilliai's	391.57	1.23	34.52	1.23	17.04	1.17
Lawley–Hotelling	676.26	1.23	183.88	1.23	66.26	1.17
Roy's	1137.97	1.29	367.89	1.29	250.28	1.29

Once it was detected that the characterization functions differ significantly according to the type of network and this factor has greater effect, using the test of

Tukey, multiple comparisons were carried out, with the objective of detecting which characterization functions differ significantly in each level of the factor of type of network.

Tukey's tests carried out showed the means of the characterization functions Avg, EL, EG and CG are significantly equal for scale free and exponential networks, and for the random networks the mean is significantly greater. For the function Std, the means of the scale free and random networks are significantly equal and for the exponential networks it is significantly smaller. The characterizations functions where significant differences in the means of the random, scale free and exponential networks were observed are L and D. Then it can be conclude, according with the cases mentioned above, that the Avg, Std, EL, EG, and CG functions are weak relevant characteristics; the L and D functions are very relevant characteristics.

The characterization functions selected to discriminate among different types of networks, were L, Std and EL, in this way a characterization function of each group representing the cases 1 and 2 were taken into account. The Avg and CG were not taken into account because of the abnormalities presented in the plots of the residuals, thus D and EG also were excluded because of being highly correlated with L, which indicates redundancy.

To obtain an efficient classifier of networks, a quadratic discriminant analysis was carried out with different combinations of the characterization functions selected, using in both cases the cross-validation procedure. In Table 3, the percentages of networks classified correctly inside each type of network, are shown and the total percentage of correct classifications.

Table 3. Results of the Quadratic Discriminant Analysis.

Quadratic Discriminant Analysis				
Combination of variables	Random Networks	Scale Free Networks	Exponential Networks	Total
L	97.8% (88 / 90)	78.9% (71 / 90)	50.0% (45 / 90)	75.6%
L, Std	97.8% (88 / 90)	88.9% (79 / 90)	95.6% (84 / 90)	94.1%
L, EL	97.8% (88 / 90)	72.2% (63 / 90)	66.7% (60 / 90)	78.9%
L, Std, EL	98.9% (89 / 90)	100% (90 / 90)	100% (90 / 90)	99.6%

4 Conclusions and Future Work

An experimental design was presented for the purpose of selecting statistically characterization functions of complex networks relevant and not redundant, the results obtained in the experimentation show that the shortest path length, the standard deviation and the local efficiency, permit in a quantitative way to discriminate among random networks, scale free networks and exponential networks. By means of the quadratic discriminant analysis an accuracy of classification of the 99.6% was obtained, the size of the sample was justified statistically. As it can be appreciated in this work the number of characterization functions is very small in relation to the 47 functions utilized in [17], this reduces dramatically the time of computation required for the classification of complex networks.

In future works it is recommended to extend this experimental design for networks with greater amount of nodes and other types of networks, besides it is recommended to work with procedures and algorithms for selection of characteristics utilized in the area of machine learning in order to compare the results obtained with this experimental design, and work with real instances of graphs as Internet.

5 References

[1] V. Latora, and M. Marchiori, "Efficient Behavior of Small World Networks", Physical Review Letters, 2001, pp. 198701-1 – 198701-4

[2] A. L. Barabási, R. Albert and H. Jeong, "Mean-Field Theory for Scale-free Random Networks", Physica A, 1999, pp. 173-189.

[3] S. Maslov, K. Sneppen, A. Zaliznyak, "Detection of Topological Patterns in Complex Networks: Correlation Profile of the Internet". Physica A, 2004, pp. 529-540.

[4] R. Albert and A. L. Barabási, "Statistical Mechanics of Complex Networks", Reviews of Modern Physics, 2002, pp. 47-97.

[5] L. A. Adamic, R. M. Lukose, A. R. Puniyani and B. A. Huberman, "Search in power law network", Physical Review E, 2001, pp. 046135-1 - 046135-8.

[6] L. Costa, F. A. Rodrigues, G. Travieso and P. R. Villas, "Characterization of Complex Networks: A survey of measurements." arXiv:cond-mat/0505185v5, 2007.

[7] B. Bollobás and O. M. Riordan, "Mathematical results on scale-free random graphs", Handbook of Graphs and Networks, Wiley-VCH, Berlin, 2002, pp. 1-32.

[8] R. Albert, H. Jeong, and A. L Barabási, "Error and attack tolerance of complex networks", Nature, 2000, pp. 5234-5237.

[9] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On power-law relationship on the internet topology", ACM SIGCOMM Computer Communication Review, 1999, pp. 251-262.

[10] A. L. Barabási, "Emergence of Scaling in Complex Networks", Handbook of Graphs and Networks, Wiley-VCH, Berlin, 2003, pp. 69-82.

[11] M. E. J. Newman, "The structure and function of complex networks", SIAM Review, 2003, pp. 167-256.

[12] L. A. N. Amaral, A. Scala, M. Barthelemy and H.E. Stanley, "Classes of small world networks", PNAS, 2000, pp. 11149-11152.

[13] P. Sen, S. Dasgupta, A. Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna, "Small-world properties of the Indian Railway network", Physical Review E, 2003.

[14] A. L. Barabási and R. Albert, "Emergence of Scaling in Random Networks", Science, 1999, pp. 509-512.

[15] Z. Liu, Y. Lai, N. Ye and P. Dasgupta, "Connectivity distribution and attack tolerance of general networks with both preferential and random attachments", Physics Letters A, 2003, pp. 337- 344.

[16] L. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection", Journal of Machine Learning Research, 2003, pp. 1157 - 1182.

[17] S. K. Singhi and H. Liu, "Feature Subset Selection Bias for Classification Learning", Proceedings of the 23rd ICML, 2006, pp. 849 – 856.

[18] E. M. Airoldi and K. M. Carley, "Sampling algorithms for pure network topologies: a study on the stability and the separability of metric embeddings" ACM SIGKDD Explorations Newsletter, 2005, pp. 13 - 22.

[19] M. Middendorf, E. Ziv, C. Adams, J. Hom, R. Koytcheff, C. Levovitz, G. Woods, L. Chen, and C. Wiggins, "Discriminative Topological Features Reveal Biological Network Mechanisms", BMC Bioinformatics 2004, 2004.

[20] E. Ziv, R. Koytcheff, M. Middendrof, and C. Wiggins, "Systematic Identification of statistically significant network measures. http://arxiv.org/PS_cache/cond-mat/pdf/0306/030 6610v3.pdf, 2005.
[21] W. Ali, R. J. Mondragón, and F. Alavi, "Extraction of topological features from communication network topological patterns using self-organizing feature maps", http://arxiv.org/abs/cs.NE/0404042, 2004.
[22] Montgomery, D.C. Diseño y Análisis de Experimentos. Limusa Wiley, 2004.

A *First* Approach to Understanding the Anomalies on My PC *

Luis M. Fernández, Hugo Terashima-Marín, Manuel Valenzuela-Rendón

{A00789695, terashima, valenzuela}@itesm.mx
Center for Intelligent Systems
Tecnológico de Monterrey, Monterrey Campus
Monterrey, Mexico

Abstract. Nowadays, and more than ever before, we live on a *computerized* world. Computers are used every day, all day, all around the world. All of people's activities now rely on PCs and, therefore, rely on the perfect execution of the applications running in these machines. As one can see, any malfunction of these machines may cause a huge problem in today's globalized world economy and safety.

This is a *first* approach to trying to understand such malfunctions (i.e., anomalies) that represent such a threat. The interaction of applications in the PCs are recorded and analyzed applying machine learning techniques, looking for possible combinations of applications that may lead to anomalies.

By applying simulation and machine learning techniques, some combinations of applications were found that, eventually, may lead to an anomalous event. Moreover, these combinations were clustered to see how similar they were, and concluding that there was a kind of defined set of combinations that may cause anomalies. Though, a simple approach is presented in this paper, the obtained results might be then used to improve the performance and profiling of everyday PCs. In this paper, all these findings are presented followed by the future work and conclusions related to this work.

1 Introduction

Computers have entered almost every arena of human society nowadays. They operate in our homes, our workplaces, and our schools. They come in so many shapes and sizes that it is sometimes difficult to recognize them: while laptop and desktop computers are commonplace, computers can also be found in home electronics, automobiles, airplanes, automatic teller machines (ATMs), security systems, and many other devices and situations. Many of the world's societies depend heavily on computers in the operation of their transportation systems, commerce, utilities, law enforcement, governance, and more [1].

From the above facts, it is easy to see that people's everyday activities rely not only on the applications that run on the PCs, but also on the correct and

* This project is funded by Tecnológico de Monterrey, Research Chair CAT 010

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 193–203

Received 26/06/07
Accepted 31/08/07
Final version 20/09/07

flawless execution and interaction of these programs. Failing to appreciate the importance of the later may cause serious problems. Just to show this, consider the power grid's computers that keep power flowing in 15 USA states and one Canadian province. More than 50 million people were left without electricity on August 14, 2003, thanks in part to a software bug [2]. Moreover, consider the 13 Root Domain Name servers, two of which are operated by VeriSign, that each day receive 25 billion requests for Web addresses. This system is a frequent target of computer malcontents, and a February 6 attack caused three of the servers to drop 90% of their queries during a 12-hour period [2].

All these reasons, and many more [3,4], have created the need to develop tools and techniques to deal with these *anomalies*. Consequently, this has given birth to *Anomaly Detection* which can be described as an alarm for strange system behavior [5]. The concept stems from a paper fundamental to the field of security - "An Intrusion Detection Model", by Dorothy Denning [6]. In it, she describes building an *activity profile* of normal usage over an interval of time. Once in place, the profile is compared against real time events. Anything that deviates from the baseline, or the norm, is logged as anomalous.

Currently, there are many approaches that deal with anomaly detection. However, most of them look for anomalies analyzing network traffic or providing algorithms to detect intruders, without considering that anomalies are also present in just one PC (i.e., a home desktop). Some examples of current research are given by [7], [8] and [9]. In this paper, a first approach to detecting anomalies when using a computer is presented, looking at the interactions among typical applications. The main objective is to find common rules that might be useful to prevent the presence of anomalies. It is not possible that, given today's computer power, people are faced, occasionally, with a frozen application and/or total use of PC power by just one application.

In the following sections of this paper, the existing difference between this work and other publications is presented, the proposed approach to this problem is explained and detailed. Next, the implementation of this research is described, followed by the discussion and analysis of the obtained results. Finally, the conclusions of this current research are presented along with the future work needed to achieve the next stages of this research.

2 Related Work

Most of anomaly detection current research is devoted to the study of network traffic analysis and intrusion detection systems, profile generation and statistics and specification-based anomaly detection [6,7,8,9]. However, the present work is different in the sense that it attempts to study the complex interactions among applications that usually run in any PC. It is believed that before trying to study complex interaction among computers (e.g., distributed systems), it is important to understand what might go wrong or what might cause a PC and its programs to trigger a problem.

Anyway, there are some projects related to collecting data associated to the applications that run on a computer. On this, one of the latest projects is the Flight Data Recorder (FDR) [10] which logs all persistent state interactions of a PC. They propose a new way to store these logs and to access such information. However, the main objective of the data they provide is to improve not only how large systems are managed, but also how to prevent attacks (which they refer to as anomalies). As such, the present work is different since it attempts, first, to discover common application interactions that may lead to an anomalous event; and, second, to study how frequent such combinations appear when using a computer.

Jude Shavlik and Mark Shavlik [11] proposed the use of machine learning techniques to study combinations in PCs. Nevertheless, and even though the idea behind is the same, the way it is used and the goals are different. Jude and Mark Shavlik aim at finding intruders in the system whereas the present paper looks for the interactions of applications that may represent an anomaly.

This is an approach to studying the combinations of applications and the possible consequences that they may represent in the performance of any PC. The next section presents the steps taken to tackle this issue.

3 Approach

The followed approach is rather simple to explain. First, it is necessary to gather data, always the hardest thing to do. Each time step, all the processes that represent some PC load (not memory allocation) are recorded; these applications are usually those in use by the user. The applications in memory are not considered since they do not represent PC load, unless they are in use and not only loaded. Next, all this data is stored in a database for further analysis. In this analysis, rules that trigger anomalies are discovered and clustered employing machine learning techniques. In this case, three simple algorithms were tried out: k-means [1] [12], filtered clustering [2] [13] and farthest first [3] [14,15]. This approach is depicted in Figure 1.

However, there is still a problem that has to be overcome; there is a lack of available data to conduct our analysis. Although, getting data from a PC should be easy, there are some barriers that prevent doing so, being *privacy* issues the most important one. Almost no one is willing to let a program record all the programs a person makes use of [16]. This situation was solved simulating a PC and the programs running on it; however, there was the need of some statistics in order to get close-to-real-life results for this paper, more precisely:

- the most popular applications used by people, and

[1] The k-means algorithm is an algorithm to cluster objects based on attributes into k partitions.

[2] An algorithm that filters possible noise and unwanted attributes.

[3] A hierarchical clustering algorithm based on a measure of dissimilarity between observations.

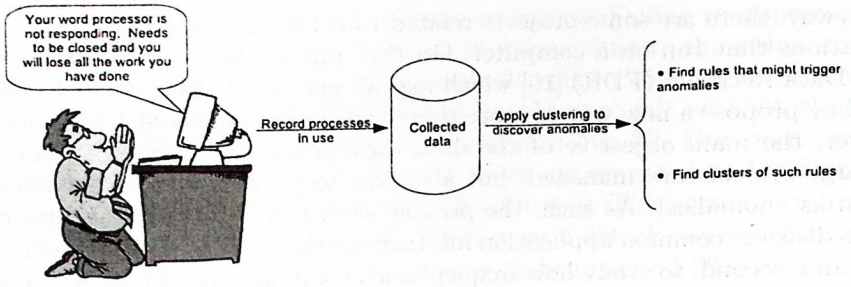


Fig. 1. Proposed approach to find anomalies.

– the percentage each of these applications represent as PC load

These data was provided by MugShot [17] where a complete list of most used applications plus their statistics and parent groups were provided. All these help simulate a PC running the most common applications and generate the required data to conduct the analysis.

Having overcome the lack of meaningful data, the next step is the implementation of this approach. This is explained and detailed in the next section of this paper.

4 Implementation

In this section the implementation of the proposed approach is presented. There are three aspects of the implementation that will be explained here: the simulation, the storage, and the machine learning tools applications.

4.1 Getting Data through Simulation

The simulation was conducted using Turtlekit [18], a MadKit plugin for multi-agent simulation. In this case, an abstract class was coded in Java, and all the applications, plus its characteristics, inherited from it. In the proposed simulation, each application is represented by an agent, instantiated with a different color, different life time, different PC load, and, for the purposes of the simulation, different position in the Turtlekit grid.

The different applications are instantiated at execution time using the statistics provided by MugShot web site [17]. Each application is associated with a probability, the applications that are used the most have a higher probability of being instantiated, whereas those that are seldom used have a smaller probability. Every new application that is placed in the Turtlekit world has its own characteristics (e.g., color, life time, etc.) This probability-instantiation approach is implemented using the genetic algorithm roulette wheel selection [19].

Figure 2 shows a moment of the simulation process that was conducted in order to get the data needed. Each little square represents a different application



Fig. 2. An example of the simulation process.

instance with a particular life time, so each application is in use for a period of time. Every time step a new application is instantiated and others *die*. Here is when the applications are recorded and stored for the next step and to do this another agent was coded. This agent looks at the grid and keeps a record of the agents alive at each stage of the simulation. This agent also keeps a sum of the percentages each in-use application represents. As long as this total sum is less than 100%, the possibility of having a *possible* anomaly is low. This is so under the assumption that most of the time computers do work well, otherwise, people would not make use of them [4]. On the other hand, once this total addition gets to 100%, then, a flag of *possible* anomaly is associated to the list of alive applications. The assumption that making use 100% of CPU resources could be a *possible* anomaly is not far from the truth. Whenever all CPU resources are being used there is a period of time when the PC simply does not respond. However, as the title suggests, this approach is a very first step towards really understanding what may cause an anomaly. There are other factors to be considered such memory paging, device usage, etc., which are not considered for the time being.

The process described above is conducted each step time, and the results are stored for further analysis. This storing process is described in the next section.

4.2 Storage

In this part of the project the applications in used are stored. These are stored as a word vector and a time marker. In other words, for each time t_i there will be a vector V_i that contains all the applications $[app_1^i, app_2^i, \cdots, app_n^i]$. The size of the vector is not constant since each time step there might be a different number of applications in use.

This last fact was the main reason why each group of applications was stored as a word vector. If these lists of applications had been stored as a key followed by the applications as possible attributes, an sparsed matrix would have been the result originating more problems than providing solutions.

[4] More on this assumption is discussed in the conclusions

4.3 Machine Learning Techniques Application

In this stage, machine learning techniques are applied to the data collected. It is important to point out that the data collected was quite huge. There were 10 runs of the simulation, where each run took 5 hours. The resulting size of the data sets after each simulation run was around 2GB.

It is clear to see that a very versatile tool was needed to conduct this stage of the research. As such, "Yet Another Learning Environment" (YALE) was used [20]. This platform provides a complete set of tools to conduct machine learning related experiments and analysis. Moreover, it is completely coded in Java which makes it easy to combine with other Java-based APIs; in this case MadKit.

Three machine learning algorithms were applied to the data [12,13] and [14,15]. All this algorithms considered the frequency of each single term present in each vector. In other words, these algorithms look for how many times an application app_i happens in each vector V_i. For the k-means algorithm [12], a random number seed of 10 was used and for farthest first [14,15], the random number seed was 1. The results gotten after using these techniques are presented in the next section.

5 Obtained Results and Discussion

All the data gathered after running the experiments was studied using machine learning algorithms. These results are presented in the next paragraphs. It is worth pointing out that in the following results, the proper names of the applications have not been included, and they will be referenced using their main use.

5.1 Mined Rules

Using the algorithms already mentioned, a set of rules was mined trying to characterize what applications may trigger a possible anomaly. In this case, only those lists of applications that were flagged as *possible* anomalies were studied.

The results gotten are shown in Table 1. It is interesting to see that the presentations utility is present in two of the four rules. Also, it can be seen that the email manager is also present in two of the mined rules. Also something quite interesting to see is that just the music player application might be enough to cause an anomaly.

Table 1. Mined rules that may trigger a possible anomaly.

Rule1:	presentations utility	and	email manager
Rule2:	word processor	and	presentations utility
Rule3:	email manager	and	document viewer
Rule4:	music player		
Total Number of Mined Rules:			4

Table 1 tells us that there are certain applications, and a combination of these, could eventually lead to an anomaly when using a PC. However, it is also important to point out that this results were gotten using a simulation process.

5.2 Formed Clusters

The word vectors were clustered according to how similar they are. In other words, how many applications they share and how many they do not. The three clustering algorithms that were used show quite similar results. Nevertheless, in one of them the number of members of one cluster is different from the number of members in others. The number of clusters in three algorithms does not change, though.

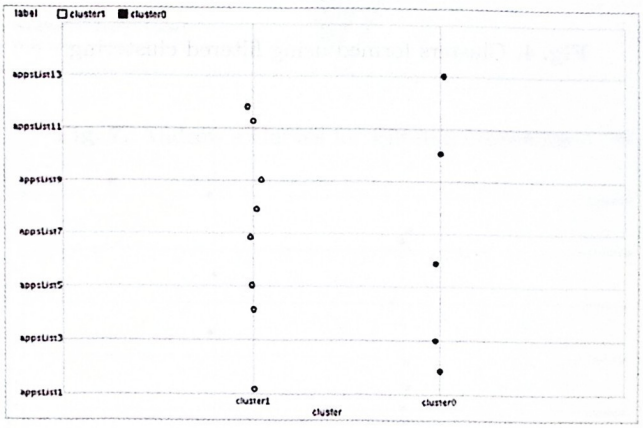


Fig. 3. Clusters formed using K-means.

Figures 3 and 4 show the obtained clusters using k-means and filtered clustering respectively. It is interesting to see that both have the same pattern and distribution on the grid.

On the other hand, Figure 5 shows the clusters using the farthest first algorithm. In this case there are two vector words that are similar to those shown in Figures 3 and 4. All these clusters just confirm that the applications in those word vectors might, indeed, lead to an anomaly when using a PC.

Furthermore, it is also important to show how *similar* this word vectors might be. In order to do so, Andrew's Curves [5] were plotted using the data provided thanks to the implementation of [21]. Figures 6, 7 and 8 show such plots.

As expected, Figures 6 and 7 show almost similar behavior and, they also show how coherent the clustering is. Figure 8 is different from the other two; however, the coherence that this graph shows is also good validating the clusters.

[5] Andrew's Curves are a useful tool for separating multivariate observation into groups that can not easily be distinguished in a tabular presentation, where curves of similar observations generally overlap, while dissimilar observations fall into different groups.

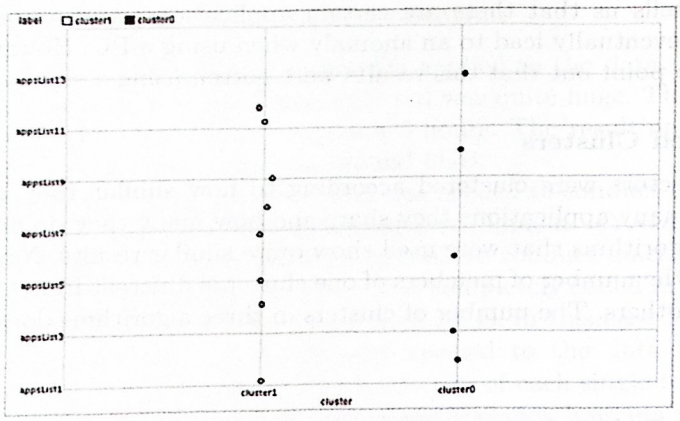


Fig. 4. Clusters formed using filtered clustering.

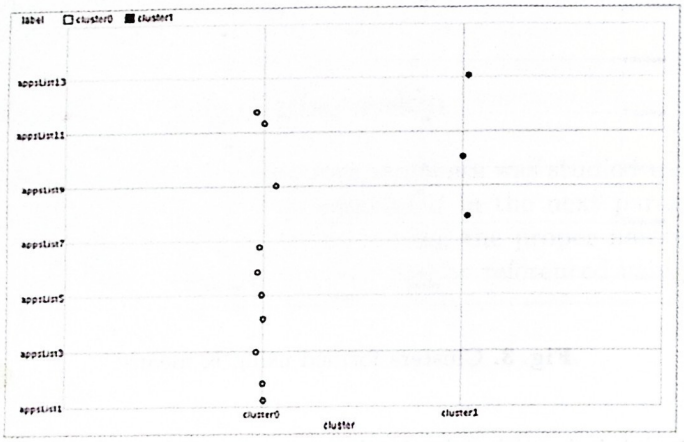


Fig. 5. Results using farthest first clustering algorithm.

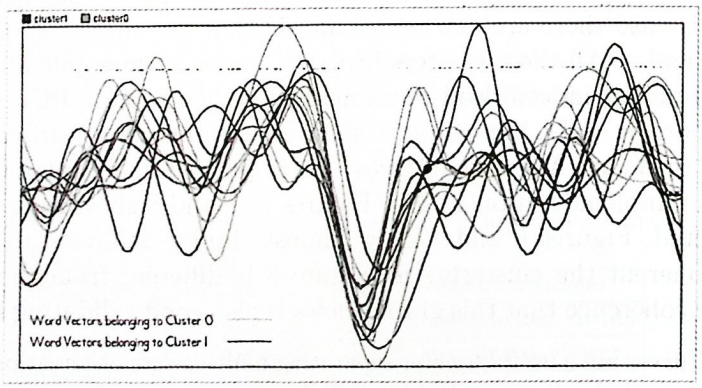


Fig. 6. Andrew's curves for K-means.

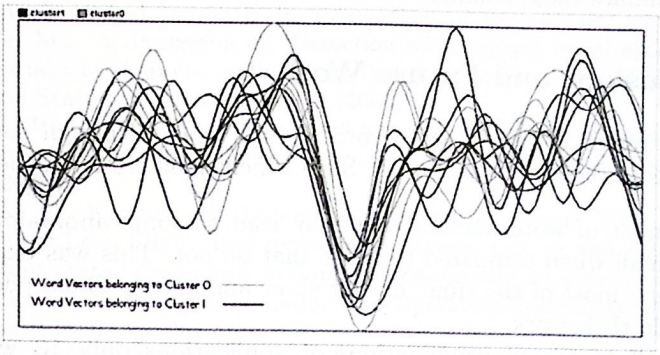


Fig. 7. Andrew's Curves for Filtered clustering.

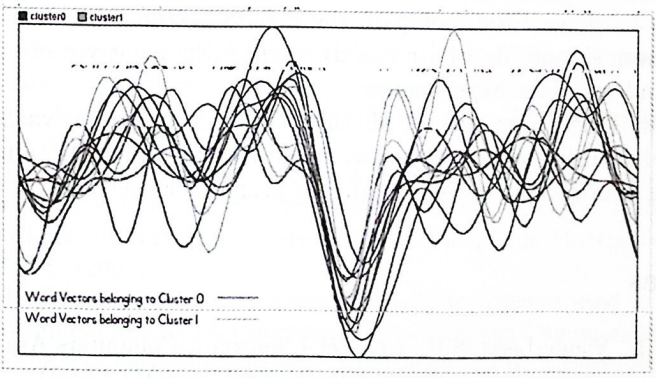


Fig. 8. Andrew's Curves for Farthest First.

All these results show that the rules found might be warnings that indicate possible anomalies and that the sets of applications that potentially lead to anomalous events do share some common characteristics. In the next section, the conclusions of the present work are reported along with the future work needed to validate these results.

6 Conclusions and Future Work

The present research has obtained some results that, although some of them were expected, others were not at all. Such conclusions are listed below.

1. The number of word vectors that may lead to some anomalous events is rather small when compared to those that do not. This was expected since computers, most of the time, do not show anomalies. Thus, people rely on PCs to do their work.
2. Among all generated combinations of applications only 13 word vectors caused an anomaly flag.
3. The applications that could potentially lead to an anomaly is also very small. The rules that were built show that only 5 of all 21 most common applications represent some threat.
4. In general, the possible combination of applications that may lead to an anomaly is the same across all seen flagged word vectors. This was shown when they clustered in a rather tight way.

Nevertheless, there are still some things left to be done, and the most important one is to go beyond the *simulation* and actually get real data. There are some approaches to such task [10]; however, they are not totally specific to the situation under studied here. Once this is done there are two other things that have to be done. First, compare the results gotten with real data versus simulation; and, second, find false positives and false negatives crossreferencing simulation and real data experiments.

Finally, it is necessary to use all this new knowledge to warn of *possible* anomalies when using a PC. One way could be to implement an entity that monitors the applications in use, building a profile per each user.

References

1. Nyhoff, J.L., VanderLeest, S.H.: Chapter 1, Lesson 1: Computers Are Everywhere. www.calvin.edu/academic/rit/webBook/chapter1/lesson1/ (2005)
2. Hesseldahl, A.: Our Fragile Computerized World. (2007)
3. Reeves, G.E., Neilson, T.C.: The Mars Rover Spirit FLASH anomaly. In: Proceedings of the Aerospace Conference, IEEE Computer Society Press (2005)
4. Associated-Press: Engineers labor to fix space computers. www.msnbc.msn.com/id/19224133 (2007)

5. Tanase, M.: One of these things is not like the others: The state of anomaly detection. http://www.securityfocus.com/infocus/1600 (2002)
6. Denning, D.: An Intrusion-Detection Model. In: Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society Press (1986)
7. Deri, L., Suin, S., Maselli, G.: Design and implementation of an anomaly detection system: An empirical approach. In: Proceedings of Terena TNC. (2003)
8. Basseville, M.: A discussion on 'Detection of intrusions in information systems by sequential change-point methods' by Tartakovsky, Rozovskii, Blažek, and Kim. Journal on Statistical Methodology **3** (2006)
9. Kim, S.S., Reddy, A.L.N.: Image-based anomaly detection technique: Algorithm, implementation and effectiveness. Journal on Selected Areas in Communications **24**(10) (2006)
10. Verbowski, C., Kiciman, E., Daniels, B., Kumar, A., Wang, Y.M., Roussev, R., Lu, S., JuhanLee: Flight data recorder: Always-on tracing and scalable analysis of persistent state interactions to improve systems and security management. In: Proceedings of the Seventh Symposium on Operating Systems Design and Implementation (OSDI). (2006) 117 – 130
11. Shavlik, J., Shavlik, M.: Selection, combination, and evaluation of effective software sensors for detecting abnormal computer usage. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. (2004) 276–285
12. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: The analysis of a simple k-means clustering algorithm. In: Symposium on Computational Geometry. (2000) 100–109
13. Witten, I., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.: Weka: Practical machine learning tools and techniques with java implementations. In: Proceedings of ICONIP/ANZIIS/ANNES'99 Int. Workshop: Emerging Knowledge Engineering and Connectionist-Based Information Systems. (1999) 192–196
14. Hochbaum, Shmoys: A best possible heuristic for the k-center problem. Mathematics of Operations Research **10**(2) (1985) 180–184
15. Dasgupta, S.: Performance guarantees for hierarchical clustering. In: 15th Annual Conference on Computational Learning Theory. (2002) 351–363
16. Web-Page: Privacy and Security (BS7799, ISO 17799, Computer Forensic). www.netlitigation.com/netlitigation/privacy.htm (2007)
17. Web-Page: Application Statistics - MugShot. http://mugshot.org/applications-learnmore (2007)
18. Michell, F.: Introduction to turtlekit: A platform for building logo based multi-agent simulations with madkit. Technical Report RR LIRMM 02215, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, Université Montpellier II (2002)
19. Blickle, T., Thiele, L.: A comparison of selection schemes used in genetic algorithms. Technical Report 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich (1995)
20. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), Springer (2006)
21. García-Osorio, C., Maudes, J., Fyfe, C.: Using andrews curves for clustering and sub-clustering self-organizing maps. In: Proceedings of the European Symposium on Artificial Neural Networks - ESANN2004. (2004) 477 – 482

Modelling and Prediction of the MXNUSD Exchange Rate Using Hybrid RLS-BP Interval Type-1 Non-Singleton Type-2 Fuzzy Logic Systems

Gerardo M Mendez[1], Angeles Hernandez[1]

Instituro Tecnologico de Nuevo Leon
Av. Eloy Cavazos #2001, CP 67170, Cd. Guadalupe, NL, México
gmm_paper@yahoo.com.mx, ahernandez@yturria.com.mx

Abstract. This paper presents a novel application of the interval type-1 non-singleton type-2 fuzzy logic system (FLS) to one step ahead prediction of the daily exchange rate between Mexican Peso and US Dollar (MXNUSD) using the recursive least-squared (RLS)-back-propagation (BP) hybrid learning method. Experiments show that the exchange rate is predictable and according to a simple short-term investment strategy, a good annual profit rate can be obtained. A non-singleton type-1 FLS and an interval type-1 non-singleton type-2 FLS, both using only BP learning method, are used as a benchmarking systems to compare the results of the hybrid interval type-1 non-singleton type-2 FLS (RLS-BP) forecaster.

Keywords: Type-2 fuzzy inference systems, type-2 neuro-fuzzy systems, hybrid learning, uncertain rule-based fuzzy logic systems.

1 Introduction

Interval type-2 fuzzy logic systems (IT2 FLS) constitute an emerging technology [1]. As in hot strip mill process [2], [3], in autonomous mobile robots [4], and in plant monitoring and diagnostics [5], [6], financial systems are characterized by high uncertainty, nonlinearity and time varying behavior [7], [8]. This makes it very difficulty to forecast financial variables such as exchange rate, closing prices of stock indexes and inflation. The ability of comprehending as well as predicting the movements of economic indices could result in significant profit, stable economic environments and careful financial planning. Neural networks are very popular in financial applications and a lot of work has been done in exchange rate and stock markets predictions described elsewhere [9]–[11].

T2 fuzzy sets let us model the effects of uncertainties, and minimize them by optimizing the parameters of the IT2 FS during a learning process [12]–[14]. Although some econometricians claim that the raw data used to train and test the FLS forecasters should not be directly used in the modelling process, since it is time varying and contains non-stationary noise, they were used by us to directly train the IT2 FLS. The inputs were modeled as type-1 (T1) non-singletons, incorporating the

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 204–213

Received 18/06/07
Accepted 31/08/07
Final version 21/09/07

uncertainties of the training data in the antecedents and consequents of the fuzzy rule base. The interval type-1 non-singleton type-2 fuzzy logic system (IT2 NSFLS-1) forecaster accounts for all of the uncertainties that are present in this application, rule uncertainties due to training with the noisy data and measurement uncertainties.

2 MXNUSD Exchange Rate Prediction

2.1 Noisy Input Output Data Pairs

The data used to train the three forecasters cover a period of nine years and four and a half months from 01/02/97 to 06/17/06 whereas the test data cover five months from 06/18/06 to 10/12/05. The daily closing price of MXNUSD exchange rate was found on the Web site: http://pacific.commerce.ubc.ca/xr/. It was a set of $N = 2452$ noisy data, $x(1)$, $x(2)$, ..., $x(2452)$. It is assumed that the noise free sampled MXNUSD exchange rate, $s(k)$, is corrupted by uniformly distributed stationary additive noise $n(k)$, so that

$$x(k) = s(k) + n(k) \qquad k = 1,2,...,2452 \ . \tag{1}$$

and that signal to noise ratio (SNR) is equal to 0 dB. Figure 1, shows the trend of the raw data. The first 2352 noisy data were used for training, and the remaining 100 data were used for testing the forecasters. Four antecedents were used as inputs, $x(k-3)$, $x(k-2)$, $x(k-1)$ and $x(k)$, to predict the output, $y = x(k+1)$.

2.2 IT2 NSFLS-1 Design

The architecture of the IT2 NSFLS-1 was established in such away that its parameters were continuously optimized. The number of rule-antecedents (inputs) was fixed to four. Each antecedent-input space was divided into three FS, resulting in 81 rules. Gaussian primary membership functions (MF) of uncertain means were chosen for the antecedents. Each rule of the IT2 NSFLS-1 was characterized by 12 antecedent MF parameters (two for left-hand and right-hand bounds of the mean and one for standard deviation, for the FS of each of the four antecedents), and two consequent parameters (one for left-hand and one for right-hand end points of the consequent type-1 FS), giving a total of 14 parameters per rule. Each input value had one standard deviation parameter, giving 4 additional parameters.

The resulting IT2 NSFLS-1 used type-1 (T1) non-singleton fuzzification, join under maximum t-conorm, meet under product t-norm, product implication, and center-of-sets type-reduction.

The IT2 NSFLS-1 was trained using two main learning mechanisms: the back-propagation (BP) method for tuning of both antecedent and consequent parameters, and the hybrid training method using recursive least-squared method (RLS) for tuning

of consequent parameters as well as the BP method for tuning of antecedent parameters. In this work, the former is referred to as IT2 NSFLS-1 (BP), and the latter as hybrid IT2 NSFLS-1 (RLS-BP).

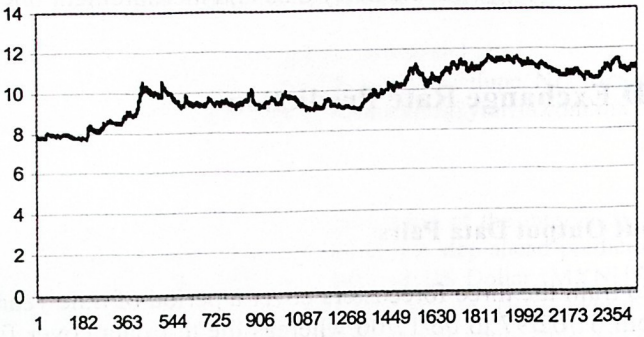


Fig. 1. The daily closing price of MXNUSD exchange rate.

2.3 IT2 Fuzzy Rule Base

The IT2 NSFLS-1 fuzzy rule base consists of a set of IF-THEN rules that represents the model of the system. This IT2 NSFLS-1 has four inputs $x_1 \in X_1$, $x_2 \in X_2, x_3 \in X_3$, $x_4 \in X_4$ and one output $y \in Y$. The rule base has $M = 81$ rules of the form

$$R^i : IF \;\; x_1 is \tilde{F}_1^i \;\; and \;\; x_2 is \tilde{F}_2^i, \;\; and \;\; x_3 is \tilde{F}_3^i \;\; and \;\; x_4 is \tilde{F}_4^i, THEN \;\; y \;\; is \;\; \tilde{G}^i . \tag{2}$$

where i = 1,2,3,...,81 rules; $\tilde{F}_1^i, \tilde{F}_2^i, \tilde{F}_3^i$ and $\tilde{F}_4^i$ are the antecedent type-2 fuzzy sets, and $\tilde{G}^i$ is the consequent type-2 fuzzy set of rule i.

2.4. Antecedent Membership Functions

The primary MF for each antecedent is an IT2 FS described by Gaussian primary MF with uncertain means:

$$\mu_k^i(x_k) = \exp\left[-\frac{1}{2}\left[\frac{x_k - m_k^i}{\sigma_k^i}\right]^2\right] . \tag{3}$$

where $m_k^i \in \left[m_{k1}^i, m_{k2}^i\right]$ is the uncertain mean, and σ_k^i is the standard deviation, with k =1,2,3,4 (the number of antecedents) and i = 1,2,...,81 (the number of M rules). Table I shows the values of the three FSs for each antecedent.

Fig. 2 shows the initial FSs of each antecedent, being the same for all inputs.

Table 1. Intervals of uncertainty for all inputs.

	m_{k1}	m_{k2}	σ_k^i
1	7.58	7.78	1.04
2	9.68	9.88	1.04
3	11.68	11.88	1.04

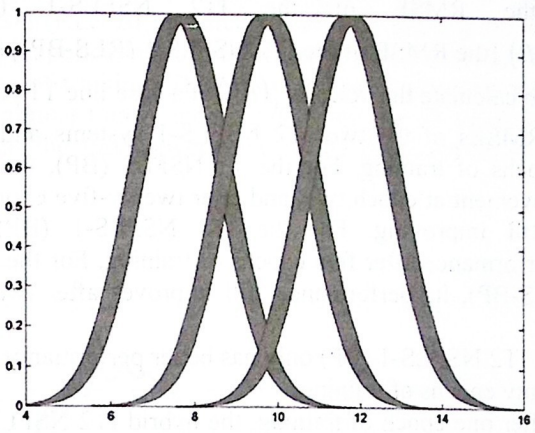


Fig. 2. Thee FSs for each of the four antecedents.

2.5. Consequent Membership Functions

The primary membership function for each consequent is a Gaussian with uncertain means, as defined in (3). Because the centre-of-sets type-reducer replaces each consequent set $\tilde{G}^i$ by its centroid, then y_l^i and y_r^i (the M left-points and right points of interval consequent centroids) are the consequent parameters.

The initial values of the centroid parameters y_l^i and y_r^i are such that the corresponding MFs uniformly cover the output space, from 4.0 to 16.0.

3 Modeling Results

One T1 NSFLS and two IT2 NSFLS-1 FLSs were trained and used to predict the daily MXNUSD. The BP training method was used to train both the base line T1 NSFLS and IT2 NSFLS-1 forecasters, while the hybrid RLS-BP method was used to train the hybrid IT2 NSFLS-1 forecaster. For each of the two training methods, BP and RLS-BP, we ran 25 epoch computations; 1138 parameters were tuned using 2350 input-output training noisy data pairs per epoch.

The performance evaluation for the learning methods was based on root mean-squared error (RMSE) benchmarking criteria as in [1]:

$$RMSE_{S,2}(*) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left[Y(k) - f_{S,2}\left(\mathbf{x}^{(k)}\right)\right]^2} \quad . \tag{4}$$

where $Y(k)$ is the output from the 100 testing noisy data pairs, $RMSE_{S,2}(*)$ stands for $RMSE_{S,2}(BP)$ [the RMSE of the IT2 NSFLS-1 (BP)], and for $RMSE_{S,2}(RLS-BP)$ [the RMSE of the IT2 NSFLS-1 (RLS-BP)]. The same formula was used in order to calculate the $RMSE_{S,1}(BP)$ of a base line T1 NSFLS (BP).

Fig. 3, shows RMSEs of the two IT2 NSFLS-1 systems and the base line T1 NSFLS for 25 epochs of training. For the T1 NSFLS (BP), there is a substantial performance improvement at epoch two, and after twenty-five epochs of training, the performance is still improving. For the IT2 NSFLS-1 (BP), there is small improvement in performance after five epochs of training. For the case of the hybrid IT2 NSFLS-1 (RLS-BP), its performance still improves after twenty-five epoch of training.

Observe that the IT2 NSFLS-1 (BP) only has better performance than T1 NSFLS-1 during the first twenty epochs of training.

In other hand, after one epoch of training, the hybrid IT2 NSFLS-1 (RLS-BP) has better performance than both T1 NSFLS (BP) and IT2 NSFLS-1 (BP) for all the twenty-five epochs. The hybrid IT2 NSFLS-1 (RLS-BP) outperforms both the T1 NSFLS (BP) and the IT2 NSFLS-1 (BP) at each epoch. The average of the $RMSE_{S,2}(BP)$ is 92.86 percent of the average of the $RMSE_{S,1}(BP)$, representing an improvement of 7.14 percent. The $RMSE_{S,2}(RLS-BP)$ improves the $RMSE_{S,1}(BP)$ by 83.52 percent and the $RMSE_{S,2}(BP)$ by 82.25 percent.

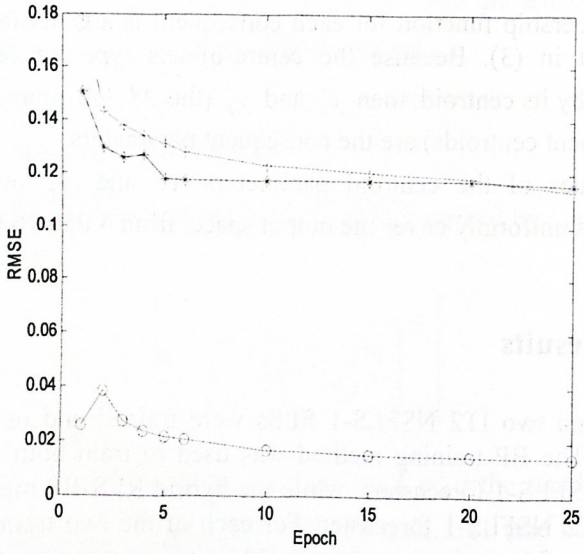


Fig. 3. TI NSFLS (BP) (+), IT2 NSFLS-1 (BP) (*), IT2 NSFLS-1 (RLS-BP) (o).

A comparison between the predicted exchange rate after one epoch and twenty-five epochs of training using T1 NSFLS (BP), and the real price are depicted in Figures 4 and 5, respectively. Figures 6 and 7 respectively show the IT2 NSFLS-1 (BP) predictions after one epoch and twenty-five epochs of training. Observe that IT2 NSFLS-1 (BP) prediction after epoch one takes better into account the variations of the real data than its counterpart T1 NSFLS-1 (B). The predicted exchange rate using hybrid IT2 NSFLS-1 (RLS-BP) after one epoch and twenty-five epochs of training, are depicted in Figures 8 and 9, respectively. After only one epoch of training, the hybrid IT2 NSFLS-1 (RLS-BP) predictions are very close to the real values, meaning that it is feasible its application to predict and to control critical systems where there is only one chance of training. The reason is that the uncertainties presented in the training data are almost totally incorporated into the 81 rules of the IT2 NSFLS-1 forecaster, and that the fast convergence of the predicted error is improved by the RLS method.

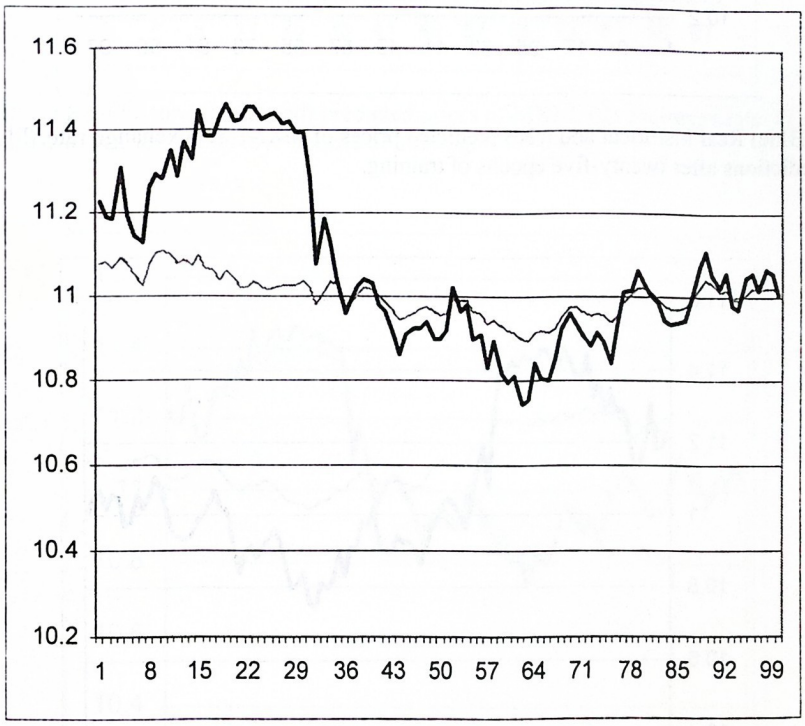


Fig. 4. (Blue) Real historical and (Red) predicted prices of MXNUSD exchange rate. T1 NSFLS (BP) predictions after one epoch of training.

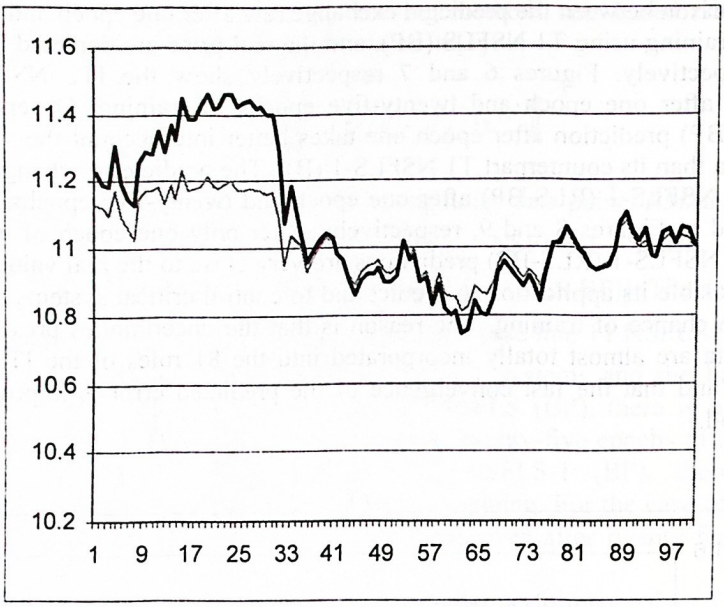


Fig. 5. (Blue) Real historical and (red) predicted prices of MXNUSD exchange rate. T1 NSFLS (BP) predictions after twenty-five epochs of training.

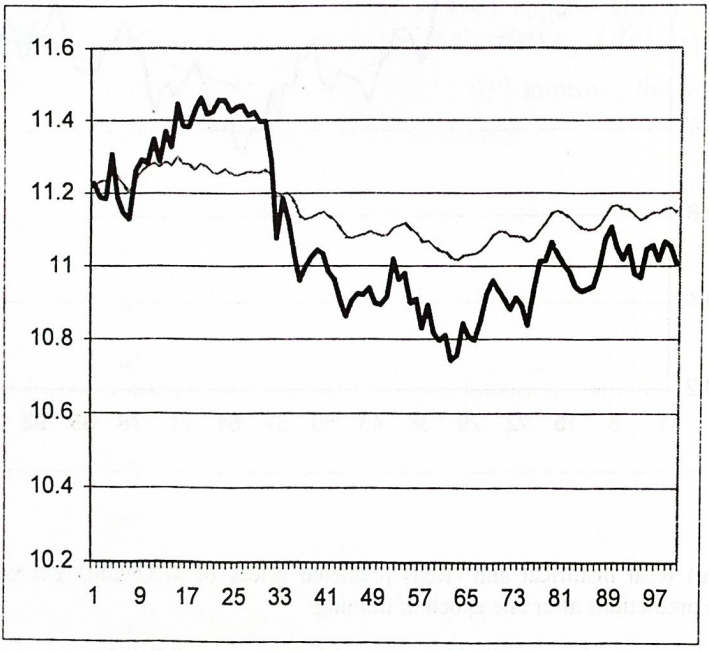


Fig. 6. (Blue) Real historical and (red) predicted prices of MXNUSD exchange rate. IT2 NSFLS-1 (BP) predictions after one epoch of training.

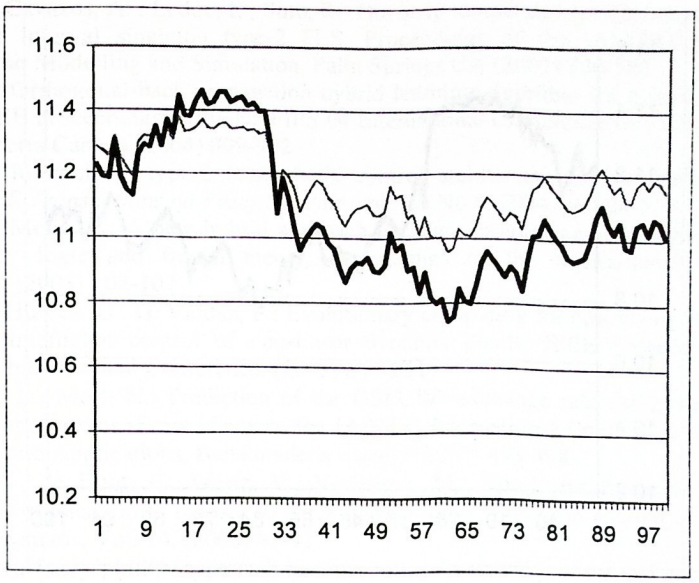


Fig. 7. (Blue) Real historical and (red) predicted prices of MXNUSD exchange rate. IT2 SFLS-1 (BP) predictions after twenty-five epochs of training.

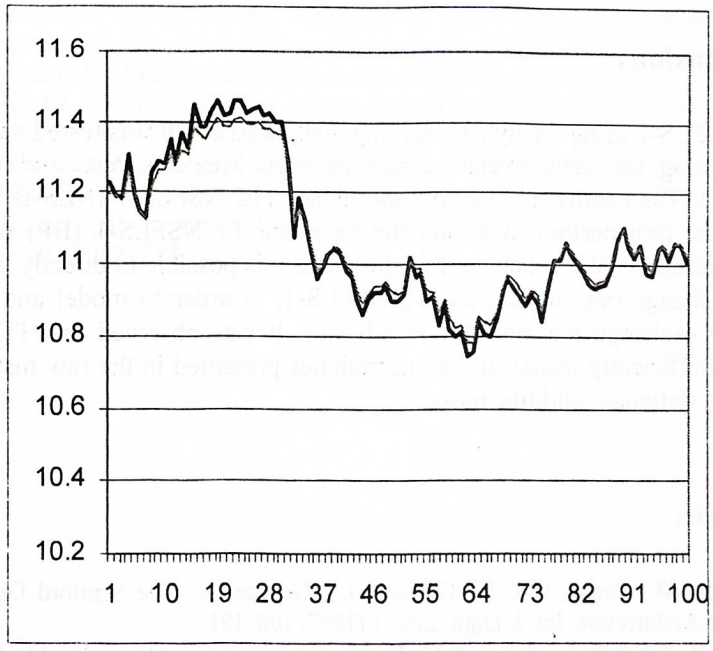


Fig. 8. (Blue) Historical and (red) predicted prices of MXNUSD exchange rate. Hybrid IT2 NSFLS-1 (RLS-BP) predictions after one epoch of training.

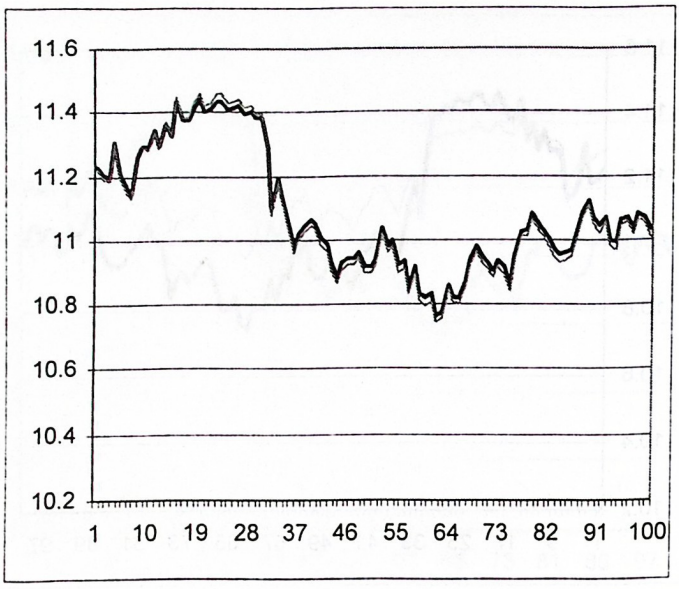


Fig. 9. (Blue) Historical and (red) predicted prices of MXNUSD exchange rate. Hybrid IT2 NSFLS-1 (RLS-BP) predictions after twenty-five epochs of training.

4 Conclusions

An IT2 NSFLS-1 using the hybrid learning method RLS-BP was tested and compared for forecasting the daily exchange rate between Mexican Peso and U.S. Dollar (MXNUSD. The results showed that the hybrid IT2 NSFLS-1 (RLS-BP) forecaster provided the best performance, and the base line T1 NSFLS-1 (BP) provided the worst performance. We conclude, therefore, that it is possible to directly use the daily data of exchange rate to train an IT2 NSFLS-1, in order to model and predict the MXNUSD exchange rate one day in advance. It was observed that IT2 NSFLS-1 forecasters efficiently managed the uncertainties presented in the raw historical data, modeled as stationary additive noise.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438
1. Mendel, J. M.: Uncertain rule-based fuzzy Logic systems: Introduction and New Directions. Upper Saddle River, NJ: Prentice-Hall, (2001)

2. Mendez, G., Cavazos, A. , Leduc, L., Soto, R.: Hot strip temperature prediction using hybrid learning for interval singleton type-2 FLS. Proceedings of the IASTED International Conference on Modelling and Simulation, Palm Springs CA (2003) 380-385
3. Mendez, G.: Orthogonal-back propagation hybrid learning algorithm for type-2 fuzzy logic systems. IEEE Proceedings of the NAFIPS 04 International Conference on Fuzzy Sets, Vol. 2. Banff Alberta Canada (2004) 899-902
4. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots", *IEEE Transactions on Fuzzy Systems*, vol. 12, No 4, 2004, pp 524-539
5. Castillo, O., Melin, P.: A new hybrid approach for plant monitoring and diagnostics using type-2 fuzzy logic and fractal theory. Proceedings of the International Conference FUZZ'2003, (2003) 102-107
6. Castillo, O., Huesca, G., G. Valdez, F.: Evolutionary computing for optimizing type-2 fuzzy systems in intelligent control of non-linear dynamic plants. IEEE Proceedings of the NAFIPS 05 International Conference, (2005) 247-251
7. Anastasakis, L., Mort, N.: Prediction of the GSPUSD exchange rate using statistical and neural network models, Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Benalmadena España (2003) 493-498
8. Lendasse, A., de Boot, E., Wertz, V., Verleysen, M.: Non-linear financial time series forecasting – application to the bel 20 stock market index. European Journal of Economics and Social Systems, Vol. 14. (2000) 81-91
9. Ito, Y.: Neuro-fuzzy phillips-type stabilization policy for stock process and exchange-rates models. Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Benalmadena España (2003) 499-504
10. Saad, E. W., Prokhorov, D. V., Wunch II, D. C.: Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEE Trans. On Neural Networks, Vol. 9. (1998) 1456-1470
11. Leigh, W., Paz, M., Purvis, R.: An analysis of a hybrid neural network and pattern recognition technique for short term increases in the NYSE composite index. The International Journal of Management Science, Vol. 30. (2002) 169-176
12. Liang, Q., Mendel, J. M.: Interval type-2 fuzzy logic systems: theory and design. Trans. on Fuzzy Systems, Vol. 8 (2000) 535-550
13. John, R. I.: Embedded interval valued type-2 fuzzy sets. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, 1 & 2, Honolulu Hawaii (2002) 1316-1321
14. Mendel, J. M., John, R. I.: Type-2 fuzzy sets made simple. IEEE Transactions on Fuzzy Systems, Vol. 10. (2002) 117-127

Entropy Based Linear Approximation Algorithm for Time Series Discretization

Acosta-Mesa Héctor-Gabriel, Cruz-Ramírez Nicandro and García-López Daniel-Alejandro

Universidad Veracruzana, Departamento de Inteligencia Artificial, Sebastian Camacho No. 5, Col. Centro C.P. 91000, Xalapa, Veracruz, México.
{heacosta, ncruz}@uv.mx, dalexgarcia@gmail.com

Abstract. Time series data mining is a relatively new sub-area of data mining, in which the temporal dimension of data introduces new challenges for classification and clustering tasks. The huge amount of information contained in temporal databases requires efficient representations, not only to reduce dimensionality, but also to preserve the relevant information for efficient classification. Many approaches have been proposed to represent temporal data in a discrete form. However, most of them are oriented to data compression, rather than to information maximization. In this work we propose new time series discretization algorithm called EBLA2. The basic idea behind EBLA2 is to minimize the entropy of the temporal patterns over their class labels after finding a minimum set of intervals from which the continuous values of the temporal database can be discretized. Under a similar approach, the algorithm is able to find the minimum time series length to represent the complete time series database.

1 Introduction

The measurement of observations that evolve over time is becoming a very common task due to the huge amount of applications that produce this kind of data; for example, in medical or industrial applications. Unfortunately, temporal data produces enormous databases that require efficient representations, not only to reduce dimensionality, but also to preserve the relevant information for efficient classification. Moreover, most of the algorithms for classification work only with discrete data [10]. Many approaches have been proposed to represent temporal data [8][6], all of them are oriented to data compression, rather than to information maximization; that is to say, those representations transform times series of length N, into a set of n coefficients, where $n < N$. These data compression processes are only intended to reduce dimensionality of the data. However, they do not take in account if the new representation preserves the relevant information to maintain the membership of the observations to the label class associated to each one. Discretization algorithms which maximize information on data will improve the efficiency of classifications tasks performed on it, and also could be used as a feature detection tools [4].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 214–224

Received 17/06/07
Accepted 31/08/07
Final version 21/09/07

In most of the algorithms proposed to discretize time series, the user requires to specify a set of parameters to perform the transformation; for example, the number of segments (word size) to divide the times series length, and the number of intervals (alphabet) required to compress the time series values. Without a previous analysis of the temporal data, it is very difficult to know the proper parameter values to obtain a good discrete representation of data. However, in practice it is assumed that the parameters are known.

In the present work, we propose a new algorithm for time series database discretization called EBLA2 (Entropy Based Linear Approximation). Under this approach the data transformation is done in a supervised fashion. The goal is to find the minimum word size and alphabet size, and the range values for each interval which minimize the entropy of the discrete representation with respect on the class label of the database. The content of the paper is presented as follows: in section 2 the related work is discussed. Section 3 describes the proposal and main concepts involved in its definition. In section 4 the algorithm EBLA2 is explained in detail. Section 5 describes the experiments carried out and the data used to evaluate the performance of the algorithm. Section 6 shows the results obtained over the experiments. Section 7 presents a discussion, and finally in section 8, conclusions and future work are presented.

2 Related work and motivation

Based on the divergence of Kullback-Leibler, in [9] a time series discretization method is proposed, the core of the algorithm is to detect the persistent states on the time series that consist of recurring persistent states produced by an underlying process, this is a unsupervised method because does not require specification of class label. The applicability of this method is restricted to the existence of persistent states in the time series, something that is not very common in most of real applications. The persistent algorithm processes a single time series at a time, then the discretization criterion is not generalized to the complete dataset.

Dimitrova et al [2], using different approach represents time series as a multi-connected graph; under this representation, similar time series are grouped into a graph model. The algorithm focuses on minimizing the number of connected nodes (graph) to represent a model time series under different criterion based on Single-Link Clustering (SLC) algorithm, plus one criterion to consider the entropy to determine which arcs on the graph to be deleted. Each node in the graph represent one point in the time series. The main drawback of this algorithm is then its computational complexity ($O(MN^4)$,where N is the number of nodes and M is the number of time series contained on the database) that makes it difficult to apply on big datasets.

More recently, an algorithm called Symbolic Aggregate Approximation (SAX) has been proposed [8]. This algorithm is based the Piecewise Aggregate Approximation representation [1]. SAX algorithm requires the user to define an alphabet (A) with fixed interval ranges and a word size (Ws) as parameters. After a nor-

malization procedure, times series length is partitioned into (Ws) segments, the corresponding values are mapped to one of A discrete values through the use of a normal probability density function (PDF). One advantage of this representation is that, once the dataset has been transformed, a smoothed version of the original data can be recovered using the PDF. Although SAX has been created for streaming data, it has proved to be an efficient representation for classification and clustering. However, as SAX works in a unsupervised fashion. it does not take advantage of class labels to improve classification performance. The present work, responds to the necessity of having an algorithm that not only discretizes temporal databases in order to reduce dimensionality, but also that automatically determines the reduction scale of the time series (word size) and the number of intervals (and its different ranges) that maximize the classification accuracy. The proposed algorithm EBLA2, takes as antecedent the discretization algorithm Class-Attribute Interdependence Maximization (CAIM) reported in [6]. CAIM algorithm works with static data in a supervised approach, the main idea of CAIM is to keep the interdependence relationship between attributes and class labels using a proposed information gain metric, called CAIM measure. In a similar way, EBLA2 proposes a metric to define an equivalent measure for time series discretization, it uses information gain measured in terms of entropy. For dimensionality reduction the algorithm takes as an antecedent the time series representation Piecewise Aggregate Approximation (PAA)[1], it consists of obtaining the mean values of a predefined segment over the time series. In the following section a complete description of the proposal is introduced.

3 Proposal

Discretization is concerned the process to map variables with continuous values into discrete values. This process has been widely used to compress data to facilitate computation in terms of space and time. More formally given the data domain $x|x \in \mathbb{R}$, where $\mathbb{R}$ is the set of reals and the discretization scheme(D) $D = \{[d_0, d1], (d_1, d_2], ..., (d_{n-1}, d_n]\}$ where d_0 and the d_n are the minimum and maximum value of x respectively. Every pair of values represents an interval, one of each maps the specific range of continuous values to one element of a discrete set {1..m}. Where m is called the discretization degree and $d_i|i = 1..n$ are the interval limits, also know as a cut points [6]. Discretization process can be split in two main jobs. The first one is to find the number of discrete groups to do the mapping from continuous to discrete. The second one is to define the range or limits of each interval in the continuous domain [6]. Both jobs are done by EBLA2 using a principle of information gain based on entropy. The aim of the algorithm is to find the number of intervals, and their limits from which the membership of the resulting discrete models are clustered with respect to the class label. As explained in the following section, the discretization scheme is computed for the whole dataset, i.e. all the time series values are considered to find the discretization scheme with minimum entropy.

Although it is almost the same process, for the sake of clarity we have divided the time series discretization in two phases . The first is to find the alphabet size and the second is to find the word size. On each phase the objective is not only to find the number of discretization values, but also the range of the interval. It can be thought of as a discretization on both axis, x and y respectively, see figure 1 . EBLA2 is based on PAA representation, which consists in obtaining the mean values of each segment in which the time series is divided.

3.1 Dimensionality reduction

A simple and efficient time series representation is Piecewise Aggregate Approximation (PAA), where each segment has equal size and the number of segments is determined for user. Often, a priori information is not available, so the goal is to find the number of intervals with different size for reduction of dimensionality at the same time as maintaining the information with respect to the class label. Let C be a time series with length n represented in a w-dimensional space (word) as a vector $\overline{C} = \overline{c_1}, ..., \overline{c}_w$. and $T = \{t_1, t_2, ..., t_w\}$ be the discretization scheme where t_i is the interval of time of i segment of $\overline{C}$. Where the i element of $\overline{C}$,is computed as:

$$\overline{c}_i = \frac{1}{|t_i|} \sum_{j=1}^{|t_i|} C_{t_j} \tag{1}$$

where: $|t_i|$ is the number of elements of t_i. Figure 1 shows a continuous time series $S_r|s_i \in \mathbb{R}, i = 1..300$ and its corresponding discrete representation $S_d|s_j, j = 1..3$ (word size =3), with $D = \{[c_0, c_1], ...(c_4, c_5]\}$ where $c_0 = -10,..,c_5 = 60$. As shown in figure 1, every single value is mapped to one out of three discrete values, according to the intervals found by the algorithm $[c_0, c_1] = 1, ..., (c_4, c_5] = 5$ (see horizontal doted lines), and intervals time $T = \{t_1, t_2, t_3\}$ where $t_1 = [1, 66]$, $t_2 = [67, 150]$, $t_3 = [151, 300]$ (see vertical doted lines)

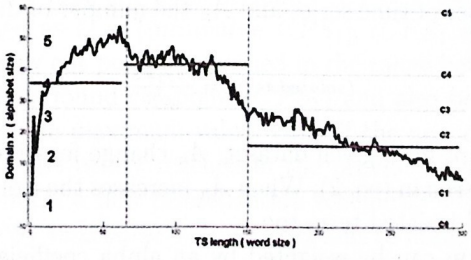


Fig. 1. Time series representation. Raw data S_r is represented in EBLA2 as S_d={4,5,2}

3.2 Utility Measure

Most discretization algorithms require heuristics to avoid the a priori definition of the alphabet size. For example, in temporal databases, the information criterion and [2], and the persistence score [9], information entropy maximization

(IEM), information gain, maximum entropy, Petterson-Niblett and Minimum Description Length (MDL) [6],[3] has been used. In the present work, the proposed criterion to select the optimal cut points is based on the information gain measure, defined for a specific discretization scheme and its corresponding class labels. Formally, the information gain based on entropy can be stated as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in A_n} \frac{\#S_v}{\#S} Entropy(S_v) \tag{2}$$

Where: S and A are two different set time series.
$A_n \subseteq S | a_i \in A_n \wedge a_i \notin (A_n \setminus \{a_i\}), i = 1..n$
$\#S_v$ is the number of time series with value v in S
$\#S$ is the number of time series in S
The entropy of S is given by:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \, log_2 \, p_i \tag{3}$$

Where: c is the number of classes.
p_i is the probability of class i in S
The entropy of S when it takes the value S_v is

$$Entropy(S_v) = \sum_{i=1}^{c} p(S|v)_i \, log_2 \, p(S|v)_i \tag{4}$$

Where: c is the number of classes.
$p(S|v)_i$ is the conditional probability of class i in S given a time series with value v.

Because it is possible to find different cut points with equal information gain, we propose to aggregate one term to the selection criterion. This new term is a heuristic to select one of the tied candidates, this term biases the selection to those cut points that generates higher different time series patterns given a discretization scheme. In other words, it prefers a discretization schemes which generate more isolated temporal patterns. This metric is computed as follows: let $\#S$ be the number of time series and A_n the number of different time series:

$$Isolated_term = \frac{\frac{A_n}{log_2 \#S}}{\#S} \tag{5}$$

$\#S$ remains constant for a given dataset. A_n change for different discretization scheme as it can be seen in (eq. 2). When A_n increases the isolated term increases and when decreased isolated term too.

The Isolated term can be weighted by an alpha coefficient to modulate its influence over the entire weight. The behavior of the isolated term in a database with 500 instances is shown in figure 2.

The goodness of a cut point is given by the sum of terms (eq. 2) and (eq. 5)

$$Utility = Gain + Isolated_term \tag{6}$$

It is important to remark that the entire time series is considered as one attribute, and that the discretization scheme is considered for the whole database. It allows the algorithm to find a good global solution, that minimizes the entropy on data.

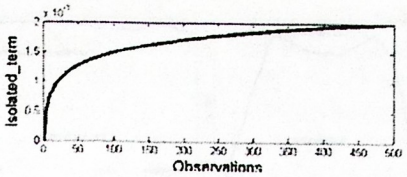


Fig. 2. Isolated term behavior. Cut points which discernes higher amounts of observations are better ranked

4 Algorithm

Under the utility measure (eq. 6) explained above, the time series discretization process can be thought of as the search on the space of all possible discretization schemes. The discretization problem on time series can be divided into two subproblems, the first one is to find the number and range of intervals of time that maintain the relevant information (alphabet), and the second is the discretization of data on each interval of time (word size). EBLA2 has been designed to solve both problems. A discretization scheme is conformed by a set of intervals defined by the cut points. In order to select each interval to be included in the discretization scheme, a set of prospectus cut points has to be created and evaluated. Some algorithms for discretization of static data, like CAIM [6], use to create the group of cut point candidates as the middle points of the different adjacent values contained in all existing continuous values (see figure 3).

To do this job, on the first iteration the algorithm searches for a cut point (between the minimum value and the maximum value contained all over the continuous range of the time series values) with maximum utility in terms of formula (6). Once the first cut point is found (CPS_1), the first discretization scheme is constructed as $D = \{[minvalue, CPS_1], (CPS_1, maxvalue]\}$. On the second iteration, a new cut point is searched in the range between the minimum value and the previous found cut point (CPS_1) and also between the previous found cut point and the maximum value. Only if the new cut point found has a bigger utility than the utility reached in the previous iteration, the new cut point is inserted to form a new discretization scheme $D = \{[minvalue, CPS_1], (CPS_1, CPS_2], (CPS_2, maxvalue]\}$. It is important to note, that on every single iteration a new set of utility values is generated. The iterative process continues until the new cut point does not produce a utility improvement. The cut point selection explained above can be extremely expensive, considering that a continuous attribute could have a huge number of different values, and that many of them could be very similar. A more efficient approach used in this work is, instead to use all the different values, to form representative values computed as percentiles. On the present algorithm we evaluate the cut points locating the percentiles defined from 0.0% to 100% on increments of 0.1%.

On the second phase of the discretization process (word size computation), the cut points represent intervals of time, and then are intended, to reduce the

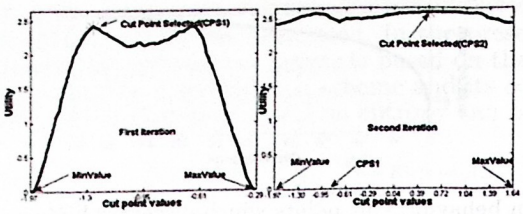


Fig. 3. Cuts point selection. (a) Cut points obtained in the first iteration $D = \{[minvalue, CPS_1], (CPS_1, maxvalue]\}$ (b) Discretization scheme obtained on the second iteration $D = \{[minvalue, CPS_1], (CPS_1, CPS_2], (CPS_2, maxvalue]\}$

time series length. This process consists in joining adjacent intervals keeping the information gain. The utility is computed in the same way as computed to find the alphabet, but in this case instead to add cut points , the objective is to delete cut points. The starting each point of time series is considered as a cut point and the job is to delete as much cut points as accessible keeping the classification accuracy reached on the previous phase (alphabet computation), a cut point is consider every single observation on the time series. Using an iterative approach, the entropy is computed for every deleted point. At the end of this cycle, the cut point with more information is deleted and the cycle continues until a new delete cut point does not improve the entropy. In this case point selection is made finding which prospect to eliminate from all prospectus cut points. This process can be though of as the elimination of those observation that does not affect the classification performance.

It is important to see, that this second phase has to be done after to find the best alphabet, having in mind to reduce the time series length preserving the relevant information on data. The pseudocode of the EBLA2 algorithm is shown below.

First phase:

$VD = sort(Unique(S_p))$
$C = Percentiles(VD)$
$MiddlePoints = \sum_{i=1}^{SizeC-1} \frac{C_i + C_{i+1}}{2}$
$B = min(C) \cup max(C)$
$GlobalUtility = 0$
$L_b = -inf$
While $L_b >$ $GlobalUtility$ and not empty(C)
    $L = \{\}$
    foreach c ∈ C and c ∉ B
        $L = L \cup FUtility(D(S, T, G(B \cup \{c\})))$
    $b = max_i(L)$ //Index of value maximum of L
    if $L_b > GlobalUtility$
        $B = B \cup c_b$
        $GlobalUtility = L_b$
    $C = C \setminus \{c_b\}$
return B

Where $S = \{s_i | s_i \in \mathbb{R}, i = 1..n\}$ represents the numeric values of a time series S with length n. S_p is the set of p time series. $T = \{t_i | i = 1..r\}$ is the temporal

scheme with r intervals of time is divided. $C = \{C_k \in \mathbb{R}, k = 1..m\}$ is the set of m prospectus cut points. G is a function for generating one discretization scheme E, given a set of limits B. $B = \{C_a | C_a \in C\}$ is a cut points set on C. $E = \{(B_{b-1}, B_b] | B_b > B_{b-1} b = 2..g+1\}$is a discretization scheme of length g. D is a discretization function to transform the set S_p into a discrete form given a discretization scheme. $FUtility$ is the utility value of D computed as described in (eq. 6). In the first phase the process starts using a discretization scheme: The value of T contains all the intervals of the complete time series.

On the first phase, the algorithm starts using a discretization scheme with a unique interval. Iteratively, the algorithm increments the number of intervals using the cut point selection procedure described above. The stop criterion was defined when the addition of a new cut point does not provide an improvement in terms of information gain. Specifically, in our implementation we used the difference of (eq. 6) at a consecutive iterations evaluated as:

$$Utility^{iteration} < Utility^{iteration-1}$$

Second phase:

$C = \{2, .., n\}$
$T = \{1, ... n\}$
$GlobalUtility = FUtility(D(S, T, G(B)))$
$L_b = inf$
While $min(L_b) >= GlobalUtility$ and not empty(C)
    L={}
    foreach c $\in$ C and c $\notin$ B
        $L = L \cup FUtility(D(S, H(T, c), G(B)))$
    end
    $b = max_i(L)$ //Index of value maximum of L
    if min(L_b) >= $GlobalUtility$ and not empty(C)
        $T = H(T, L_b)$
return T

In the second phase, the algorithm joins the consecutive bits of the time series in the following way : Let $H(P, p)$ be a function that joins (with the left interval of p) two adjacent intervals of p in P . It is important to remember that the value of B (alphabet) was computed in the previous phase.

5 Experiments

The performance of the algorithm was tested using twenty databases available on the time series classification/clustering WEB page (www.cs.ucr.edu/~eamonn/time_series_data), see five first columns in table 1.

Different time series representations have been evaluated using these databases [?]. It is important to remark that most of these representations have been developed to compress data, rather that to improve classification performance. One of the most efficient representation recently proposed is Symbolic Aggregate Approximation (SAX)[8]. Although SAX representation has been designed to discretize time series for streaming data purposes, it has shows a good performance on classification and clustering tasks. Because as far as we know at the time at which this work was developed there was no a representation specifically designed to optimize classification performance, the performance of EBLA2 was compared to those results obtained by SAX.

Table 1. Databases properties, alphabet and word sizes obtained for EBLA2

DataSet	Num. Classes	Size training	Size testing	Length TS	Word Size	Alphabet Size
50Words	50	450	455	270	40	2
Adiac	37	390	391	176	90	9
Beef	5	30	30	470	23	3
CBF	3	30	900	128	64	2
Coffee	2	28	28	286	75	3
ECG200	2	100	100	96	48	2
Fish	7	175	175	463	119	3
Face(All)	14	560	1690	131	33	3
Face(Four)	4	24	88	350	45	2
Gun Point	2	50	150	150	24	2
Lighting 2	2	60	61	637	80	2
Lighting 7	7	70	73	319	81	2
Osu Leaf	6	200	242	427	54	2
Olive Oil	4	30	30	570	287	3
Swedish Leaf	15	500	625	128	67	5
Trace	4	100	100	275	144	3
Two Pattern	4	1000	4000	128	64	2
Control Chart	6	300	300	60	30	3
Wafer	2	1000	6174	152	91	15
Yoga	2	300	3000	426	119	3

5.1 Classification

One nearest neighbor (1NN) was used as a classification method. Error rate was computed using Leave-One-Out Cross Validation (LOO). The discretization scheme was obtained over the training set of data. The similarity measure used in 1-NN, to evaluate the EBLA2 algorithm was the Euclidean distance. Classification performance was also evaluated using the raw data (continuous time series). It is important to remark to SAX uses its own similarity measure [8]. SAX similarity measure is defined as: Let $\hat{Q}$ and $\hat{C}$ be the SAX discrete representation of time series Q and C. Let w be the word size. *dist* function is a specific defined similarity measure defined by SAX to decode the SAX representation [8].

$$MINDIST(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (dist(\hat{q}_i, \hat{c}_i))^2} \tag{7}$$

6 Results

Results of the data reduction alphabet and word size obtained automatically after to applying EBLA2 to 20 database are shown in the 6th and 7th column of table 1. The results obtained showed that EBLA2 representation needs smaller alphabets that SAX representation to obtain similar error rates. Figure 4 shows error rates obtained on the twenty database, on the graph is observed that most of the cases EBLA2 reached error rate smaller than those obtained by SAX using the same parameters (alphabet size and word size).

7 Discussion

Generally speaking, error rates reached using EBLA2 representation were smaller that those obtained by SAX tested using the same parameters (word size and alphabet size), however higher alphabet sizes on SAX obtains smaller error rates than EBLA2.

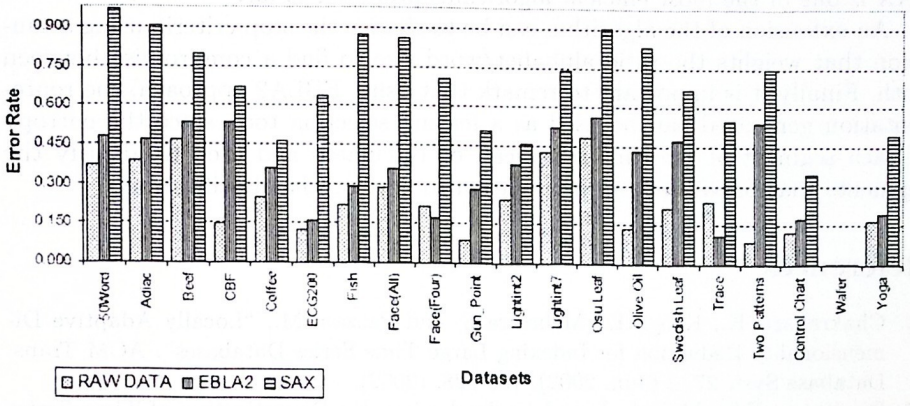


Fig. 4. Error rate obtain for EBLA2 and SAX using the parameters suggested by EBLA2. Error rate reached using raw data is shown as well

This seems to indicate that EBLA2 finds a local minimum derived of its greedy approach. In order to solve this, aleatory movements to be required to implement at the beginning of the search. The proposed discretization approach used by EBLA2 is an efficient technique to automatically compute the parameter for discretization(alphabet and word size), reaching competitive error rates. This characteristic makes EBLA2 very useful because some times it is very difficult to know a priori a good combination of alphabet and word size for a given dataset.

8 Conclusions and future work

In the present work a new algorithm for supervised discretization on time series data called EBLA2, has been proposed. Given a labeled dataset containing temporal data, EBLA2 algorithm automatically computes not only the alphabet size but also the limits which define the continuous intervals of each alphabet letter. EBLA2 also computes the number and sizes of the intervals to define the time series length. The approach proposed in this work is to evaluate different discretization schemes trying to minimize their associated entropy with respect to the class labels.

The efficiency of the algorithm was evaluated using twenty databases and compared to one of the more efficient time series discretization algorithm called

SAX and raw data. Generally, error rates reached using EBLA2 representation were smaller that those obtained by SAX tested using the same parameters (alphabet and word size). The advantage that EBLA2 does not requires a priori parameter because they are automatically calculated.

Although SAX was not developed for supervised discretization, we decided to compare EBLA2 with SAX because at the time of this publication, there was no algorithm available for supervised discretization of time series, and because SAX is one of the most efficient algorithms proposed so far.

An extension of the algorithm can be to change the stop criteria using a function that weights the ratio alphabet/word size to find a compromise between both. Finally it is important to remark that using EBLA2 approach, the representation generated can be used as a feature selection tool, since the entropy of each segment of the time series can be calculated and used to identify the segments that provides more information to develop the classification.

References

1. Chakrabarti K., Keogh E., Mehrotra S. and Pazzani M., "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases". ACM Trans. Database Syst. 27, 2 (Jun. 2002), 188-228, (2002).
2. Dimitrova, E.S., McGee, J. and Laubenbacher, E.: Discretization of Time Series Data, eprint arXiv:q-bio/0505028, (2005).
3. Fayyad U. and Irani K., "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning", In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1027, Chambery, France, Morgan Kaufmann, (1993).
4. Geurts P. , "Pattern Extraction for Time Series Classification", In Proceedings of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery(September 03 - 05, 2001). L. D. Raedt and A. Siebes, Eds. Lecture Notes In Computer Science, vol. 2168. Springer-Verlag, London, 115-127, (2001).
5. Han, J. and Kamber, M.: *Data mining, Concepts and techniques* Morgan Kaufmann, (2001).
6. Kurgan L. and Cios K., "CAIM Discretization Algorithm", IEEE Transactions On Knowledge And Data Engineering, 16(2), 145-153, (2004).
7. Last M., Kandel A., Bunke H. *Data mining in time series databases*, World Scientific Pub Co Inc, (2004).
8. Lin J., Keogh E., Lonardi S. and Chiu B., "A symbolic representation of time series, with implications for streaming algorithms", In Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (San Diego, California, June 13 - 13, 2003). DMKD '03. ACM Press, New York, NY, 2-11, (2003).
9. Mörchen F. and Ultsch A., "Optimizing Time Series Discretization for Knowledge Discovery", en Proc. of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining (Chicago, Illinois, USA, August 21 - 24, 2005). KDD '05. ACM Press, New York, NY, 660-665, (2005).
10. Tan P., Steinbach M., Kumar V.: *Introduction to data mining*, Addison-Wesley, (2006).

Constraint Programming, Planning, and Scheduling

Process Calculi for Adaptive Enumeration Strategies in Constraint Programming

Eric Monfroy[1,2], Carlos Olarte[3], and Camilo Rueda[4]

[1] Universidad Técnica Federico Santa María, Valparaíso, Chile
Eric.Monfroy@inf.utfsm.cl
[2] LINA, Université de Nantes, France Eric.Monfroy@univ-nantes.fr
[3] LIX, École Polytechnique Paris, France carlos.olarte@lix.polytechnique.fr
[4] Pontificia Universidad Javeriana, Cali, Colombia crueda@cic.puj.edu.co

Abstract. Constraint programming (CP) has been extensively used to solve a wide variety of problems. Solving a constraint problem mainly consists in two phases: propagation to prune the search space, and enumeration to choose a variable and one of its values for branching. Enumeration strategies are crucial for resolution performances. We propose a framework to model adaptive enumeration strategies using a stochastic, non-deterministic timed concurrent constraint calculus. Using the reactivity of the calculus we can design enumeration strategies that adapt themselves according to information issued from the resolution process and from external solvers such as an incomplete solver. The experimental results show the effectiveness of our approach.

1 Introduction

Constraint Satisfaction Problems (CSP) appears nowadays as a very convenient way to model various industrial applications (e.g. scheduling, timetabling, Boolean satisfiability, etc.). CSP are represented by a set of constraints (relations) between variables. Each variable is associated to a domain which represents the values the variable could be given. The resolution process consists in assigning to each variable a value from its domain such that the constraints are satisfied.

Constraint propagation based solvers are one of the most common methods for solving CSPs. This technique is complete by interleaving enumerations and constraint propagations. Constraint propagation prunes the search tree by eliminating values that cannot participate in any solution. Enumeration creates one branch of the search tree by instantiating a variable ($x = v$) and another branch ($x \neq v$) for backtracking when the first branch does not contain any solution. Although all enumeration strategies that preserve solution sets are valid, they have drastic impacts on efficiency (up to several orders of magnitude). Moreover, no strategy is the best for all problems. The issue is thus to select the right value of the right variable for enumeration. This problem can be tackled with various approaches: static and generic strategies, problem specific strategies [4], dynamic [6] or adaptive strategies to predict [3] or repair strategies [2, 5].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 227–237

Received 17/06/07
Accepted 31/08/07
Final version 30/09/07

We are interested in modeling adaptive strategies (both for repairing and predicting) using a stochastic, non-deterministic concurrent constraint process calculus, the sntcc calculus [12], an extension of ntcc[11]. The advantages of using sntcc are the following. First, we obtain a clear, formal, and homogeneous (using templates) design of the strategies. Then, the reactive aspect of sntcc is used to describe the dynamicity of the strategies, i.e. the ability of adapting, changing, or improving a strategy during the solving process. Additionally, reactivity allows us to introduce on-line expert-users or external solvers knowledge to guide the search. The stochastic aspects of the calculus enables us to rank the strategies with some probabilities of being applied: the higher the probability, the higher the strategy is judged to be efficient. The non-determinism of sntcc is used to tie-break strategies that have equal probabilities; we thus introduce randomization from which the solving process can benefit (e.g. [7]).

Using sntcc we show several types of adaptive strategies: strategies to diversify enumeration; strategies to reward "good" enumeration strategies (increasing their probabilities of being applied) (see [5]); strategies that use an incomplete but very fast solver (e.g. local search) to identify promising branches leading to a hybrid resolution mechanism; and finally strategies to non-deterministically choose a variable when several variables have the same domain size (see [7]).

This mechanism has been implemented in the Oz language [9] running a simulator of sntcc. The experimental results we obtained are more than promising. The adaptive dynamic strategies behave better than the fixed strategies.

The rest of the paper is organized as follows. In Section 2 we give an overview about constraint programming and related work regarding dynamic enumeration strategies. Section 3 presents the sntcc calculus a non-deterministic and stochastic extension of tcc [14]. Section 4 describes our framework, a generic process for describing strategies, and four instantiations of the framework. Section 5 shows some experimentations and Section 6 concludes the paper and gives some research directions.

2 Constraint propagation-based solvers

In constraint programming (CP), problems are modeled by means of a set V of variables, a set of domains for these variables D and a set of constraints C over the variables. The idea is to choose a value from D for each variable in V s.t. all the constraints are satisfied. Solvers based on constraint propagation alternate pruning of the search tree and split of the search space. The former prunes variable domains by eliminating values that cannot be part of a solution. Nevertheless, propagation is not a complete mechanism. In some cases, the propagation phase never can find a solution nor determine if there are none. In this case a split phase is required, creating two or more subproblems to continue the search. Searching for a solution in CP leads to a search tree where each node represents a new subproblem.

Although all enumeration strategies preserving solutions are valid, they drastically affect the time required to find solutions by several orders of magnitude

(see, e.g. [5] for some examples). Thus it is crucial to select a good one (that unfortunately cannot be predicted in the general case) or to eliminate a bad one (by observing and evaluating its behavior). Numerous works were conducted about split strategies. Some studies focused on generic criteria (e.g. minimum domain) for variable selection, and for value selection (e.g. lower bound, or bisection). Some works define strategies for some specific problems (e.g. [4]) where the "best" strategy can be determined according to some static criteria. However, an a priori decision concerning a good variable and value selection is very difficult (and almost impossible in the general case) since strategy effects are rather unpredictable. Information issued from the solving process can also be used to determine the strategy (e.g. [3]). [2] proposes adaptive constraint satisfaction: algorithms that behaves poorly are dynamically changed by the next candidate in a list. In [7], when several choices are ranked equally by the fixed enumeration strategy, randomisation is applied for tie-breaking. In [5], by observing and evaluating the solving process, bad strategies are eliminated whereas good ones are given more chance to be applied.

3 sntcc Calculus

Process calculi are mathematical formalisms to model and verify concurrent systems. With (few) operators, they express a wide variety of behaviors such as mobility [13], time and reactivity [14, 11], stochastic and probabilistic choices [8] among others. We are here interested in calculus derived from the Concurrent Constraint model (cc) [15]. cc is based on the concept of constraint as an entity carrying partial information, i.e. conditions on the values variables can take. This model has been extended with the notion of time in `tcc` ([14]), non-determinism and asynchrony in `ntcc` ([11]) and stochastic and probabilistic behavior in `pcc` and `sntcc` ([8, 12]).

In cc, process interactions are determined by the constraints accumulated in a global store (i.e. a set of variables and a conjunction of constraints). The store is used by processes to share information and for synchronization purposes. The store is monotonically refined by adding information using *tell* operations. Additionally, it is possible to test if a constraint c can be entailed from the store by means of so-called *ask* operations. Ask processes allows for synchronisation since they remain blocked until more information is available to entail the query.

`sntcc`[12] extends `ntcc` to model stochastic behavior. In both of them, time is conceptually divided into *time-units*. In a particular time-unit, a deterministic concurrent constraint process receives a stimulus (a constraint) from the environment and it is executed with this stimulus as the initial store. When no further evolution is possible, it responds to the environment with the resulting store. This also determines a residual process which is then executed in the next time-unit (see **next** operator below).

The model of stimulus and responses in `tcc` languages allows to model reactive systems in which agents are in permanent interaction with the environment. In our particular case, the environment will be the processes solving a CSP and

stimulus will be statistics taken from this solving process (e.g. number of variable instantiated, depth in the search tree, etc). In turns, strategies modeled with the calculus will respond to the environment with a possible enumeration strategy according to the previous knowledge and rules defined in the strategy.

3.1 Syntax and Processes Description

In this section we present the syntax of the sntcc calculus and the intuitive description of the constructs. See [12] for a full treatment of the operational semantics and the logic associated to the calculus.
Process in sntcc are built from the following grammar:

$$P, Q = \textbf{tell}\ c \mid \sum_{i \in I} \textbf{when}\ c_i\ \textbf{do}\ P_i \mid P||Q \mid \textbf{local}\ x\ \textbf{in}\ P \mid \textbf{next}\ P \mid \textbf{unless}\ c\ \textbf{next}\ P \mid\ !P \mid\ P +_\rho Q \mid\ p(\boldsymbol{x})$$

Process **tell** c adds the constraint c to the current store, thus making c available to other processes in the current time interval. Process $\sum_{j \in J} \textbf{when}\ c_j\ \textbf{do}\ P_j$ where J is a finite set of indexes, represents a process that non-deterministically choose a process P_j s.t c_j is entailed by the current store. The chosen alternative, if any, precludes the others. If no choice is possible in the current time unit, all the alternatives are precluded from execution. We shall use $\sum_{j \in J} P_j$ when the guards are **true** ("blind-choice") and we omit "$\sum_{j \in J}$" when J is a singleton. Process $P||Q$ represents the parallel composition of P and Q. Process **local** x **in** P behaves like P except that the variable x is local in P, i.e., the environment cannot see the information that P produces on x.

The only move of **next** P is a unit-delay for the activation of P. We use $\textbf{next}^n(P)$ as an abbreviation for $\textbf{next}(\textbf{next}(\ldots(\textbf{next}\,P)\ldots))$, where **next** is repeated n times. **unless** c **next** P executes the process P in the next time-unit if the current store cannot entail the constraint c.

$!P$ stands for the replication of P, i.e. the execution of a copy of P in each time-unit. This construct is equivalent to $P||\textbf{next}\ P||\textbf{next}\ ^2P....$

$P +_\rho Q$ introduces stochastic behavior. This process evolves to P with a probability ρ and to Q with a probability $1 - \rho$. This probabilistic choice can be extended to multiple processes of the form $P_1 +_{\rho_1} \ldots +_{\rho_{n-1}} P_n$ normalizing the probabilities to guarantee that $\sum p_i = 1$

Recursion in sntcc is defined by means of *processes definitions* of the form

$$p(x_1, .., x_n) \stackrel{\text{def}}{=} A$$

$p(y_1, ..., y_n)$ is an *invocation* and intuitively the body of the process definition (i.e. A) is executed replacing the formal parameter $\boldsymbol{x}$ by the actual parameters $\boldsymbol{y}$. When $|\boldsymbol{x}| = 0$, we shall omit the parenthesis.

To avoid non-terminating sequences of internal reductions (i.e., non-terminating computation within a time interval), recursive calls must be guarded in the context of **next** (see [11] for further details).

4 A framework for modelling adaptive strategies

Devising a framework for dynamic enumeration strategies resembles the idea of reactive systems [1]. Resolution processes can be viewed as the environment adding new information (stimulus) to the system to produce possibly "better" enumeration strategies. Here we propose modelling reactive and adaptive enumeration strategies using sntcc. Stimulus will be statistics taken from the solvers (e.g. depth of the search tree, number of variable instantiated, etc.) and the response of the system (resulting store in each time-unit) will be the variable (*var*) and the value (*val*) for enumeration. Constructs in the calculus will help us to define rules to choose the "*best*" strategy in each node of the search tree.

Defining dynamic enumeration strategies as sntcc processes has several advantages, namely 1) The stimulus-response model allows to clearly define the interaction between the resolution process and the dynamic enumeration strategies. 2) Complex synchronization patterns can be defined (due to blocking asks) allowing us to coordinate different processes adding new information to guide the search. In particular, we show how we can implement enumeration strategies guided by external solvers such as local search. In this case the framework can be viewed as a solver coordinator. 3) The stochastic component of the calculus allows us to rank strategies with some probabilities of being applied. Additionally, probabilities of each strategy can be dynamically changed during the solving process. 4) Non-determinism can be used to tie-break strategies that are equally ranked. Finally, 5) the parallel operator allows us to integrate incrementally new processes telling information useful to adapt the strategy or combine rules in a compositional way.

Given a CSP P with a set of constraints C and a dynamic strategy S (i.e a sntcc process) the framework works as follows: 1) The constraint solver creates the first node of the search tree imposing the constraints in C. 2) The enumeration procedure creates a new sntcc time-unit. When doing that, it feeds in the sntcc store with statistical information of the solving process as stimulus. 3) A sntcc interpreter computes the resulting store w.r.t. the stimulus and the strategy S. 4) After stability in the sntcc store, the variable (*var*) that must be split and the value (*val*) that it must take are entailed from the sntcc store. 5) Finally the enumeration procedure creates as usual two choices in the constraint solver: $var = val$ and $var \neq val$ leading to a new phase of propagation. In this way, the search tree is explored using dynamic choices according to S and statistics taken from the resolution process.[5]

A template of a strategy S in our framework can be defined as the parallel composition of three processes:

$$\textbf{Strategy} = \textbf{Choice}||\textbf{Update_Probabilities}||\textbf{External_knowledge}$$

[5] sntcc constraints and store must not be confused with constraints C in the CSP. The former are used to determine the evolution of the strategy and the latter define the problem to be solved

Update_Probabilities (**U_P**) changes the probability of each strategy according to user defined rules. Expert-user knowledge or results taken from external solvers (i.e. local search, genetic algorithms, etc.) can be used to adapt the strategy during its execution by defining an **External_knowledge** (**E_K**) process. Additionally, one could add on-line new information (constraints) as stimulus to fix some parameters in the strategy. This can be useful when expert users observing the solving process can give some hints for improving the search. Finally, **Choice** makes a probabilistic or non-deterministic choice according to information inferred from both **U_P** and **E_K**. In the following, we instantiate this generic process to define some strategies that will be tested in Section 5.

4.1 Diversifying Enumeration Strategies

Our first dynamic strategy consists in applying alternatively three different static value selection: *min* (selects the lower bound of the domain), *max* (the upper bound) and *mid* (the closest element to the middle of the domain) leading to a diversification of the value selection. The variable to be split is always the variable with the smallest domain (i.e the variable selection is fixed). The behavior of this strategy can be captured by the following process:

$$\begin{aligned}\textbf{Choice} \equiv\ & \textbf{tell}\ (val = min)\ ||\ \textbf{next tell}\ (val = mid)\ ||\\ & \textbf{next}^2\ \textbf{tell}\ (val = max)\ ||\ \textbf{next}^3\ \textbf{Choice}\\ \mathbf{S_1} \equiv\ & \textbf{Choice}\end{aligned}$$

4.2 Probabilistic Choice of the Value Selection

Although S_1 dynamically changes the value selection, it always applies the same pattern of choices. The following strategy changes the probability of applying a value selection according to the results obtained previously. The number of variables that has not been instantiated yet is the stimulus introduced by the constraint solver. The idea is to increase the probability of some value selection when it allows good pruning in the next node of the search tree. If the selection leads to a failed node in the search tree, its probability is decreased giving more chance to other strategies to be applied.

$$\begin{aligned}\textbf{Init} &= \textbf{tell}\ (\rho Min = \tfrac{1}{3} \wedge \rho Mid = \tfrac{1}{3} \wedge \rho Max = \tfrac{1}{3})\\ \textbf{U_P} &= \textbf{when}\ VarNI < VarNI'\ \textbf{do}\ increase(val') +\\ &\quad\ \textbf{when}\ VarNI \geq VarNI'\ \textbf{do}\ decrease(val')\\ \textbf{Choice} &= \textbf{tell}\ (val = min)\ +_{\rho Min}\ \textbf{tell}\ (val = mid)\ +_{\rho Mid}\ \textbf{tell}\ (val = max)\\ \mathbf{S_2} &= \textbf{Init}\ ||\ !\textbf{Choice}||\ !\textbf{next}\ (\textbf{U_P}\ ))\end{aligned}$$

In the expression above, $VarNI$ (resp. $VarNI'$) represents the number of variables not instantiated in the current (resp. previous) node of the search tree. ρMin, ρMid and ρMax are the probabilities of choosing each value selection. *increase* (resp. *decrease*) increases (resp. decreases) in a constant factor the probability of the previous value selection (val') updating the rest of probabilities to guarantee $\rho Min + \rho Mid + \rho Max = 1$. Once **U_P** has determined the

new probabilities, **Choice** makes a probabilistic choice between each alternative. Thus, the behavior of this strategy is to apply (probabilistically) the value selection that has not failed lately.

4.3 Local search based decisions

Integrating complete and incomplete methods for constraint solving has been shown promising for reducing the time required to obtain solutions in CP [10]. In our case, we use a descent algorithm for *local search* to determine the "best" value selection to continue the search in the CP solver. Let $V = \{v_1, ..., v_n\}$ be the set of variables of the CSP with domains $D = \{d_1, ..., d_n\}$. The process $LS(v_i, value, cv)$ computes local search with a neighborhood function $NB : D \rightarrow 2^D$ s.t. for all configurations c and for all $c_i \in NB(c)$, $c_i \downarrow_{v_i} = value$ i.e, the assignment $v_i = value$ is preserved during the search. In our tests, NB computes a set of configurations by changing one variable not yet determined (i.e. $|dom(v_i)| > 1$) at a time. Additionally, when no moves improve the current configuration, a random restart is performed introducing diversification in the algorithm. After a fixed number of iterations, the LS process binds cv to the number of constraints violated in the best solution found. In this way, synchronization with the rest of the system is achieved. Thus, LS processes such as $LS(v_i, min(v_i), ls_{min})$, $LS(v_i, mid(v_i), ls_{mid})$ and $LS(v_i, max(v_i), ls_{max})$ can be executed in parallel to determine the most promising branch to search for a solution (min, mid, and max being the usual value selections).

Since computing local search in each time-unit (node in the search tree) may cause overloads, we can control when the strategy uses LS-based decision. For example, we can apply LS only in the 3 first levels of the search tree (i.e $depth \leq 3$) or when more than a certain number of variables are instantiated. This kind of control can be achieved modifying the predicate *cond* in S_3:

$$\begin{aligned}
\textbf{E_K} &= LS(v_i, min(v_i), ls_{min}) || LS(v_i, mid(v_i), ls_{mid}) || LS(v_i, max(v_i), ls_{max}) \\
\textbf{Choose} &= \textbf{when } ls_{min} \leq ls_{mid} \wedge ls_{min} \leq ls_{max} \textbf{ do tell } (val = min) \\
&\quad + \textbf{when } ls_{mid} \leq ls_{min} \wedge ls_{mid} \leq ls_{max} \textbf{ do tell } (val = mid) \\
&\quad + \textbf{when } ls_{max} \leq ls_{min} \wedge ls_{max} \leq ls_{mid} \textbf{ do tell } (val = max) \\
\textbf{S}_3 &= !(\textbf{when } cond \textbf{ do } (\textbf{local } ls_{min}, ls_{mid}, ls_{max} \textbf{ in E_K} || \textbf{Choose}) + \\
&\quad \textbf{when } \neg cond \textbf{ do } S_{1/2})
\end{aligned}$$

Note that when S_3 does not apply local search based-decisions, it behaves like S_1 or S_2.

S_3 can be improved by evaluating more than one variable with LS to find the most promising branch. This can be done extending **E_K** and **Choose** processes as follows:

$$\begin{aligned}
\textbf{E_K} &= \Pi_{vi \in V_{Selected}} (LS(v_i, min(v_i), ls_{min_i}) || LS(v_i, mid(v_i), ls_{mid_i}) || \\
&\qquad LS(v_i, max(v_i), ls_{max_i})) \\
\textbf{Choose} &= \textbf{local } MinCost \textbf{ in tell } MinCost = Min(ls_{min_{1..n}}, ls_{mid_{1..n}}, ls_{max_{1..n}}) || \\
&\quad \sum_{i \in 1..n} (\textbf{when } ls_{min_i} = MinCost \textbf{ do tell } (var = v_i \wedge val = min) + \\
&\qquad \textbf{when } ls_{mid_i} = MinCost \textbf{ do tell } (var = v_i \wedge val = mid)) \\
&\qquad \textbf{when } ls_{max_i} = MinCost \textbf{ do tell } (var = v_i \wedge val = max))
\end{aligned}$$

where II stands for parallel composition of several processes and $V_{Selected}$ denotes the set of variables to be evaluated. **E_K** thus launches several LS processes that are synchronized with the non-deterministic choice in **Choose**. In this way, the variable and the value for that variable with minimum cost w.r.t. to the local search algorithms is chosen.

4.4 Randomized Variable Selection

This strategy aims at randomly (non-deterministically) choosing a variable when multiple variables have the same domain size. Hence, we are able to tie-break strategies with the same probability of being applied. This mechanism has shown good performances for some CSPs in [7].

To implement this strategy, the state of the variables in the solver (i.e. their domains) must be sent as a stimulus. In the `sntcc` store we thus have "a copy" of the variables of the solver. This strategy computes the minimal size (min_size) of the variables domains to filter the variables whose size is equal to min_size. This new list of variables is stored in a local variable $VSet$. Next, the strategy chooses non-deterministically one variable in the set $VSet$:

$$\begin{aligned}
\textbf{U_P} &= \textbf{tell}\ (Min = min_domain(Problem_vars))\ ||\\
&\quad\ \textbf{tell}\ (VSet = filter(Problem_vars, Min))\\
\textbf{Choose} &= \textstyle\sum_{v_i \in VSet} \textbf{tell}\ (Var = v_i)\\
\textbf{S}_4 &= !\ \textbf{local}\ Min, VSet\ \textbf{in}\ (\textbf{Choose}\ ||\ \textbf{Update})
\end{aligned}$$

Note that S_4 only performs variable selection. Executing it in parallel with some of the previously defined strategies leads to a scheme of dynamic variable and value selection.

5 Experimental Results

Tests presented in this section were performed in a Pentium 4 1.80GHz CPU running Linux Gentoo, kernel 2.6.15 and Oz system 1.3.1.

We run different instances of the canonical problems *Magic Sequence*, *Knights*, *Magic Square* and *N-Queens* (see [9] for an Oz program solving these problems). For each problem we compare static strategies with dynamic strategies in Tables 1, 2, 3 and 4 [6] respectively. We measure the average number of nodes explored (including failed nodes) and the average time to find one solution. In what follows, we present a brief description of each problem and the results we obtained.

Magic Square: The Magic Square Puzzle consists in finding a $N \times N$ matrix such that: 1) Every field of the matrix is an integer between 1 and N^2. 2) The fields of the matrix are pairwise distinct. 3) The sums of the rows, columns, and the two main diagonals are all equal.

[6] "-" denotes more than 1.000.000 nodes.

For this problem we compare the static value selections *min*, *mid* and *max* provided by Oz selecting always the variable with the smallest domain. Strategy S_3 executes S_1 after applying LS-based decision in the first 3 levels of the search tree and evaluates the two variables with shortest domain in each case.

Note that for instances where $N \geq 4$, dynamic strategies outperform static value selections. Furthermore, when the difference between the number of nodes explored is not significative (i.e $N < 4$), execution time is better with the static strategies due to the overload of the execution of the `sntcc` interpreter.

Knights: The idea is to find a sequence of knights moves on a $N \times N$ chessboard such that each field is visited exactly once and that the moves return the knight to the starting field, which is fixed to the lower left field.

For this problem we compare also static value selection *min*, *mid* and *max* vs dynamic strategies S_1, S_2. In this case S_1 and S_2 slightly outperform *Min* regarding the number of nodes explored and show better performances w.r.t. *Max* and *Mid*.

Magic Sequence: A magic sequence of length n is a sequence $x_0, x_1, \ldots x_{n_1}$ of integers such that for every $0 \leq i < n - 1$: 1) x_i is an integer between 0 and $n - 1$ and 2) the number i occurs exactly x_i times in the sequence.

Similar to the *Knights* problem, we compare the same strategies. According to the form of solution for this problem where the first value in the sequence must be a value close to the maximum (representing the number of zeros in the sequence) and the next one must be a small one, S_1 turns out to be a good strategy solving all the instances in 4 nodes. As expected, if it starts selecting the minimum or the mid value, we obtain results similar to the static strategies.

N-Queens: This problem consists in placing N queens on an $N \times N$ chess board such that no two queens attack each other.

We compare S_4 with the following fixed strategies: *naive* (selects the leftmost variable), *size* (selects the leftmost variable whose domain is minimal), *min* (the leftmost variable whose lower bound is minimal) and *max* (the leftmost variable whose upper bound is maximal). In all the cases the value selected is the closest to the middle of the domain (*mid*). Note that the *size* strategy is known for this problem as a good strategy exploring a few amount of nodes before finding a solution. S_4 makes some improvements randomizing the variable selection when multiple variables have the same domain size. Notice also that *size* is better regarding the execution time due to the overload of the `sntcc` interpreter in S_4.

6 Concluding remarks

We proposed a generic framework to model and integrate adaptive enumeration strategies in a constraint propagation-based solver. This framework is based on `sntcc`, a stochastic non-deterministic concurrent constraint calculus. The reactivity of the calculus allows the strategies to adapt themselves according to observations of the resolution. Additionally, parallel composition enables us to add new processes to guide the search, even external processes such as incomplete

	Max		Mid		Min		S_1		S_2		S_3	
N	Nodes	Time(s)	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
3	16	0.10	6	0.09	27	0.09	9	0.05	6.05	0.35	8.67	0.49
4	1209	0.13	380	0.16	480	0.19	3449	19.45	1611.1	8.16	101.56	1.70
5	-	-	151211	11.11	158006	10.4	573	3.10	489.10	2.51	387.5	4.01
6	-	-	-	-			2929	15.60	1585.38	8.04	528.27	6.93
7	-	-	-	-			2026	11.03	2783.87	14.06	203.25	14.62
8	-	-	-	-			257	1.77	2267.27	11.90	495.55	13.33

Table 1. Magic Square Problem

	Max		Mid		Min		S_1		S_2	
N	Nodes	Time(s)	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
8	34	0.43	46	0.40	44	0.39	33	0.82	35.07	0.84
10	1838	2.41	69	0.43	76	0.44	61	1.26	69.71	1.36
12	98	0.56	-	-	117	0.56	95	1.83	110.5	2.06
14	1116	3.09	-	-	174	0.81	142	2.83	146.63	2.9

Table 2. Knights Problem

methods (e.g. local search) or heuristics from expert-users. The framework was instantiated with 4 dynamic strategies that showed better performances than static strategies to solve four well known CSPs.

The main advantage in using this framework is that one can express complex enumeration strategies only adding/modifying *sntcc* processes in the strategy definition. Issues concerning synchronization, choices, etc. are relayed to the formal behavior provided by the calculus. Furthermore combining different enumeration strategies can be easily achieved by using parallel composition or probabilistic/non-deterministic choices. In this way, strategies can be improved running them in parallel. Finally, an interesting feature is that external solvers can be synchronized with the constraint-based solving process. Thus, information provided by them can guide the variable/value selection during the search.

Our interest in this framework is twofold: 1) to integrate more indicators from the solving process to better guide the search, and 2) to integrate more external solvers and achieve their coordination using process calculi. The final idea is to provide a generic and adaptive framework for hybrid resolution of CSPs.

References

1. G. Berry and G. Gonthier. The esterel synchronous programming language: Design, semantics, implementation. *Science of Computer Programming*, 1992.
2. J. Borrett, E. Tsang, and N. Walsh. Adaptive constraint satisfaction: The quickest first principle. In *Proc. of ECAI'1996*, pages 160–164. John Wiley and Sons, 1996.
3. T. Carchrae and J. C. Beck. Low-Knowledge Algorithm Control. In *Proceedings of the National Conference on Artificial Intelligence, AAAI 2004*, pages 49–54, 2004.
4. Y. Caseau and F. Laburthe. Improved clp scheduling with task intervals. In *Proceedings of ICLP'1994*, pages 369–383. MIT Press, 1994.

	Max		*Mid*		*Min*		S_1		S_2	
N	Nodes	Time(s)	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
70	31	0.41	35	0.39	35	0.39	4	0.42	35	0.83
90	41	0.48	45	0.41	45	0.40	4	0.42	45	0.99
110	51	0.58	55	0.42	55	0.42	4	0.42	55	1.18
130	61	0.74	65	0.47	65	0.45	4	0.45	65	1.41
150	71	0.98	75	0.52	75	0.51	4	0.45	75	1.64

Table 3. Magic Sequence Problem

	Naive		*Size*		*Max*		*Min*		S_4	
N	Nodes	Time(s)	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
32	119636	4.1	79	0.32	-	-	-	-	73.15	0.49
64	-	-	89	0.4	-	-	-	-	71.7	0.67
128	-	-	127	0.42	-	-	-	-	126.45	1.36
256	-	-	257	0.45	-	-	-	-	254.3	4.01

Table 4. N-queens Problem

5. C. Castro, E. Monfroy, C. Figueroa, and R. Meneses. An approach for dynamic split strategies in constraint solving. In *Proceedings of MICAI'05*, volume 3789 of *LNCS*, pages 162–174. Springer, 2005.
6. P. Flener, B. Hnich, and Z. Kiziltan. A meta-heuristic for subset problems. In *Proceedings of PADL'2001*, volume 1990 of *LNCS*, pages 274–287. Springer, 2001.
7. C. Gomes, B. Selman, and H. Kautz. Boosting combinatorial search through randomization. In *Proceedings of AAAI'98*, pages 431–437, Madison, Wisconsin, 1998.
8. V. Gupta, R. Jagadeesan, and V. A. Saraswat. Probabilistic concurrent constraint programming. In *Proc. of Int. Conf. on Concurrency Theory*, pages 243–257, 1997.
9. S. Haridi and N. Franzn. *Tutorial of Oz.*, 2004. Available at `www.mozart-oz.org`.
10. E. Monfroy, F. Saubion, and T. Lambert. Hybrid csp solving. In *Proceeding of FroCoS'05*, volume 3717 of *LNCS*, pages 138–167. Springer, 2005. invited paper.
11. M. Nielsen, C. Palamidessi, and F. D. Valencia. Temporal concurrent constraint programming: Denotation, logic and applications. In *Special Issue of Selected Papers from EXPRESS'01, Nordic Journal of Computing*, 2001.
12. C. Olarte and C. Rueda. A stochastic non-deterministic temporal concurrent constraint calculus. In *Proceedings of SCCC'05*, 2005.
13. J. P. R. Milner and D. Walker. A calculus of mobile processes, Parts I and II. *Journal of Information and Computation*, 100:1–77, Sept. 1992.
14. V. Saraswat, R. Jagadeesan, and V. Gupta. Fundation of timed concurrent constraint programming. In *IEEE Symp. on Logic in Comp. Science*. IEEE, 1994.
15. V. Saraswat, M. Rinard, and P. Panangaden. Semantic foundations of concurrent constraint programming. In *Proc. of POPL'91*, pages 333–353. ACM Press, 1991.

Variable and Value Selection Heuristics: Application to Solve Puzzles

Broderick Crawford[1,2] and Mary Aranda[1]

[1] Pontificia Universidad Católica de Valparaíso, Chile
[2] Universidad Técnica Federico Santa María, Chile
broderick.crawford@ucv.cl
mary.aranda.c@mail.ucv.cl

Abstract. Constraint Programming is one of the major contributions of Computer Science for solving problems of high complexity. This problems can be modeled like Constraint Satisfaction Problems and solving them via Constraint Propagation and Enumeration. In this work, we study the resolution of puzzles (Latin Square, Magic Square and Sudoku)with this approach and we evaluate the performance of different variable and value selection heuristics in the Enumeration phase, demonstrating the relevance of the Enumeration Strategies in a Constraint Programming process.

1 Introduction

The Constraint Programming (CP) has been defined as a technology of Software used to describe and solve combinatorial problems [2, 3]. The main idea of this paradigm is to model a problem by mean of a declaration of variables and constraints and to find solutions that satisfy all the constraints. Many of the combinatorial problems focused by CP can be modeled like a Constraint Satisfaction Problem (CSP), which consist of a sequence of variables $X = x_1, x_2, ..., x_n$ with its respective domains $D = D_{x_1}, D_{x_2}, ..., D_{x_n}$, and a finite set C of constraints restricting the values that the variables can take simultaneously [19]. The goal is to assign a value to each variable satisfying all the constraints. The most general notation for CSP is the following [21]: $< C; x_1 \in D_{x_1}, ..., x_n \in D_{x_n} >$

The basic mechanism underlying CP to solve a CSP interleaves Constraint Propagation (network consistency) and Enumeration (distribution or labeling) [1]. In essence, the algorithm increases the efficiency of the search by looking ahead actively using the constraints to prune the search space. Furthermore, an optimization approach is feasible from constraint satisfaction in a form of branch and bound. That is, as soon as a solution is found, a further constraint is added forcing future values to be better than the value just found according to the optimization criteria. This causes the system to backtrack until a better solution is found. When no further solutions can be found the optimum value is reached.

The underlying structure to the described paradigm CP is shown in Figure 1.

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 238–248

Received 17/06/07
Accepted 31/08/07
Final version 27/09/07

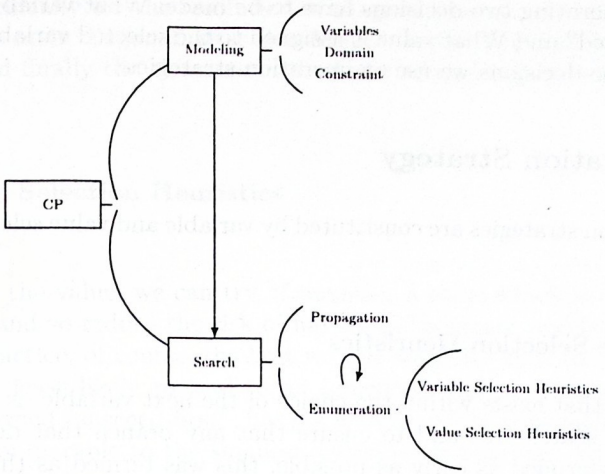


Fig. 1. Constraint Programming Structure

This work is focused on the Enumeration phase of CP, where the use of variable and value selection heuristics is critical. A suitable definition and the use of an enumeration strategy can improve the resolution process strongly. We apply to resolution of puzzles (Magic Square, Latin Square and Sudoku) different variables and values selection heuristics presented in the literature [13, 17, 4, 6].

This paper is organized as follows: in Section 2 is described the resolution technique, Section 3 and 4 show the enumeration strategies and the modeling of each problem in the platform used: Mozart[1]. In Section 5, we analyze the results obtained and finally we conclude in Section 6.

2 Resolution Technique

In the resolution of Constraint Satisfaction Problems diverse techniques can be used, currently they are solved using complete techniques (global optimization), incomplete techniques (local optimization), and hibridizations of both techniques [8]. Specifically, the Constraint Programming community uses a complete approach alternating phases of constraint propagation and enumeration [7], where the propagation prunes the search tree by eliminating values that can not participate in a solution. Enumeration [1] consists of dividing the original CSP in two smaller CSPs, creating one branch by instantiating a varible $(x = v)$ and another branch $(x \neq v)$ for backtracking when the first branch does not contain any solution.

[1] www.mozart-oz.org

When enumerating two decisions have to be made: What variable is selected to be instantiated? and What value is assigned to the selected variable?. In order to support these decisions we use enumeration strategies.

3 Enumeration Strategy

The enumeration strategies are constituted by variable and value selection heuristics [12].

3.1 Variable Selection Heuristics

The main idea that exists within the choice of the next variable, is to minimize the size of the search tree and to ensure that any branch that does not lead to a solution is pruned as early as possible, this was termed as the "fail-first" principle by Haralick and Elliot [9], described as *"To succeed, try first where you are most likely to fail"* [16, 1].

For variable selection, it is possible to find a classification according to the moment when the selection order is done.

Static Selection Heuristic: it generates a fixed order of the variables before initiating the search. Here the variables are always selected in the order predefined for instantiation.

Dynamic Selection Heuristic: it can change the instantiation order of the variables dynamically as one advances in the tree search. It is based on information generated during the search.

Here the dynamic and static terms are used according to the definition given in [2], also corresponding to the used in [16]. This definition differs from the concept used in [7, 12], where the idea of dynamism is based on adaptive constraint satisfaction [5], where the idea of dynamic selection heuristic consists basically in detecting the bad decisions made with respect to the selection of variables and values, replacing on-the-fly the strategies with bad results by others that promise to be better.

In this work we used the following variable selection heuristics:

Minimum Domain Size (*MiD*): at each enumeration step the domain of each one of the variables not yet instantiated is analyzed, then the variable with smaller domain size is selected.
Maximum Domain Size (*MaD*): the idea of this heuristic is similar to the previous one, nevertheless in this case it selects the variable with the greater domain size.
Order Previously Established (*Opre*): in this static heuristic, we fix a variable order before initiating the resolution process. The heuristic *Opre* has been

defined specifically for the resolution of magical squares, selecting first the elements of the main diagonal, after the elements in the upper triangle of the main diagonal and finally the elements in the lower triangle of the main diagonal.

3.2 Value Selection Heuristics

In choosing the value, we can try, if possible, a value which is likely to lead to a solution, and so reduce the risk of having to backtrack and try an alternative value. In practice, of course, the best we can normally do is to choose the value which seem least likely to lead to an inmediate failure. This principle, which might be termed "succeed-first", has not lead to widely-applicable value ordering heuristics comparable to the smallest-domain-first heuristic, but can give good heuristics tailored to individual problems, or types of problem [16]. In synthesis, this principle (usually problem dependent) indicates that the value with a high number of supports is likely to be preferred.

In this work we used the following value selection heuristics:

Smaller Value of the Domain (*SVal*): this heuristic establishes that the smallest value of the domain is always chosen.

Greater Value of the Domain (*GVal*): it is similar to the previous one, but instead of choosing the smallest element of the domain, the greater element is selected.

Average Value of the Domain (*AVal*): this heuristic selects the value of the domain that is more near to the half of the domain, it calculates the arithmetic average between the limits (superior and inferior) of the domain of the selected variable and in case of having a tie the smallest value is selected. In order to understand better this concept: considering that x_1 has been selected to the variable whose domain is 2#4, to select the value to instantiate calculates the average between the limits of the domain, this is:$(2 + 4)/2 = 3$, therefore the value that is selected in this stage corresponds to 3. If the domain of x_1 went 1#4 the value to select he would be 2, since the average is $(1 + 4)/2 = 2.5$ and in case of tie always the smaller value is selected.

Immediately Greater Value to the Average Value of the Domain (*GAV*): this heuristics selects the smaller value of the domain that it is greater as well to the average value of the domain. Consider the previous example, when the domain of the variable is 2#4 in heuristic the previous one selects value 3, however with present the heuristic one selects value 4. Thus, when the domain of the variable is 1#4 present the heuristic selects value 3.

Finally, established the heuristic to use, the enumeration strategies are compound according to Table 1.

S_1 = MiD+SVal	S_5 = MaD+SVal	S_9 = Opre+SVal
S_2 = MiD+GVal	S_6 = MaD+GVal	S_{10} = Opre+GVal
S_3 = MiD+AVal	S_7 = MaD+AVal	S_{11} = Opre+AVal
S_4 = MiD+GAV	S_8 = MaD+GAV	S_{12} = Opre+GAV

Table 1. Enumeration Strategies

4 Description of Problems

4.1 Magic Square

This puzzle consists in finding for a given N an NxN matrix such that every cell of the matrix is a number between 1 and N^2, all the cells of the matrix must to be different, and the sum of the rows, columns, and the two diagonals are all equal.

The mathematical model used for its representation defines a variable x_{ij} that represents the value that each cell of the matrix can take, and a variable S for the sum of each row, column and diagonal. Then the CP model establishes the following constraint:

$$\forall i,j \in \{1,...,N\} \;\; Alldifferent\{x_{ij}\} \tag{1}$$

$$\sum_{j=1}^{N} x_{ij} = S \;\; \forall i \in \{1,...,N\} \tag{2}$$

$$\sum_{i=1}^{N} x_{ij} = S \;\; \forall j \in \{1,...,N\} \tag{3}$$

$$\sum_{i=1}^{N} x_{ii} = S \tag{4}$$

$$\sum_{i=1}^{N} x_{i(N-i+1)} = S \tag{5}$$

The constraints (2) and (3) ensure that the sum of each row and each column will be equal to S, and the constraints (4) and (5) assure that the sum of each diagonal will be equal to S.

4.2 Latin Square

A Latin Square puzzle of order N is defined as an NxN matrix where all its elements are numbers between 1 and N with the property that each one of the N numbers appear exactly once in each row and exactly once in each column of the matrix.

The mathematical representation used to model the problem has a variable x_{ij} that represents the value of the cell (i, j) of the matrix. The CP model consists of the following constraints:

$$\forall i \in \{1, ..., N\} \quad Alldifferent\{x_{i1}, x_{i2}, ..., x_{iN}\} \tag{6}$$

$$\forall j \in \{1, ..., N\} \quad Alldifferent\{x_{1j}, x_{2j}, ..., x_{Nj}\} \tag{7}$$

4.3 Sudoku

Sudoku is a puzzle played in a 9x9 matrix (standard sudoku) which, at the beginning, is partially full. This matrix is composed of 3x3 submatrices denominated "regions". The task is to complete the empty cells so that each column, row and region contain numbers from 1 to 9 exactly once [10, 11].

The model used for the representation can be seen like a composition of the models used in the above puzzles, the variable x_{ij} represents the value that each cell (i, j) can take (in this case, from 1 to 9). In order to restrict that each row and each column have values from 1 to 9 exactly once the following constraint are due to impose:

$$\forall i \in \{1, ..., 9\} \quad Alldifferent\{x_{i1}, x_{i2}, ..., x_{i9}\} \tag{8}$$

$$\forall j \in \{1, ..., 9\} \quad Alldifferent\{x_{1j}, x_{2j}, ..., x_{9j}\} \tag{9}$$

On the other hand, each cell in regions S_{kl} with $0 \leq k, l \leq 2$ must be different, which forces to include in the model the following constraint:

$$\forall i, j \quad Alldifferent\{x_{ij}, x_{i(j+1)}, x_{i(j+2)}, x_{(i+1)j}, x_{(i+1)(j+1)}, x_{(i+1)(j+2)}, \tag{10}$$

$$x_{(i+2)j}, x_{(i+2)(j+1)}, x_{(i+2)(j+2)}\}$$

con $i = k * 3 + 1$ y $j = l * 3 + 1$.

5 Analysis of Results

Each one of the exposed problems were solved using the eight first strategies listed in Table 1, and additionally the strategies $S_9, ..., S_{12}$ were used in the resolution of magic squares puzzles, because these strategies are constituted by a variable selection heuristic designed specifically for such problem.

The tests conducted allow to evaluate the performance of the enumeration strategies based on the following indicators of performance:

Number of Backtracking (B): it shows the amount of bad decisions made during the search of the solution, that is calculations or decisions executed without leading to a solution.

Number of Enumerations (E): this metric tells the amount of nodes or spaces generated to find the solution of the problem, including the good enumerations that lead to a solution and the bad enumerations that force to backtrack.

Time (t): it measures the required time to solve the problem.

Each execution had a time limited to 10 minutes, not finding results are indicated with the symbol "-".

5.1 Searching the First Solution

In general it is possible to appreciate that for small instances the strategies do not reflect significant differences. Nevertheless when observing the results obtained for each one of the problems it is possible to see that the performance of the strategies varies as N increases, this because with the increase of N the size of the search space grows drastically [20]. On the other hand, when observing the results obtained it is perceived that the strategies constituted by the heuristic *MiD* $(S_1, ..., S_4)$ have better behavior in those instances in which the search space grows, this in comparison with strategies that are guided by the heuristic *MaD* $(S_5, ..., S_8)$. Such differences happen mainly because the heuristic *MiD* leads as rapidly as possible to an insolvent space, allowing to prune the tree search. Leading to an insolvent space quickly consists of choosing variables with few elements in its domain, increasing the probability of failing before generating a big search tree.

N	3			4			5			10			15		
	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)
S_1	3	0	9	6	0	10	10	0	10	67	1	18	165	7	76
S_2	3	0	10	6	0	10	10	0	10	67	1	18	165	7	78
S_3	3	0	10	6	0	10	12	0	12	70	2	18	163	5	67
S_4	3	0	10	6	0	10	10	0	11	70	4	31	309	138	229
S_5	3	0	12	8	0	10	94	77	16	-	-	-	-	-	-
S_6	3	0	9	8	0	10	94	77	15	-	-	-	-	-	-
S_7	3	0	10	8	0	10	1644	1625	106	-	-	-	-	-	-
S_8	3	0	10	8	0	10	36	21	12	-	-	-	-	-	-

Table 2. Latin Squares: Enumerations (E), Backtracking (B), CPU time (t) in ms.

Observing the results obtained for the magic square problem, one of the aspects to emphasize is the good performance obtained by the strategies constituted by the heuristic *Opre*, particularly using it in combination with value selection heuristic *AVal* and *GAV*, this good behavior reflected in the Table 3, it is due mainly because the heuristic *Opre* generates a more effective constraint propagation, reducing the size of the search space considerably. The reduction

N^2	9			16			25			49		
	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)
S_1	4	1	1	741	717	30	208861	208827	12669	-	-	-
S_2	13	7	1	1681	1655	63	9113251	9113227	505231	-	-	-
S_3	8	5	1	2465	2449	81	2187	2158	76	101510	101424	4415
S_4	13	7	1	406	384	27	7061	7037	452	7139	7081	487
S_5	16	7	1	45097	45043	3680	-	-	-	-	-	-
S_6	14	8	1	681	646	46	-	-	-	-	-	-
S_7	22	10	1	847094	847055	48519	-	-	-	-	-	-
S_8	23	19	1	358775	358772	27115	557090	557608	44189	-	-	-
S_9	5	2	0	49	34	3	26475	26428	1396	-	-	-
S_{10}	13	7	1	508	491	24	80761	80728	5078	-	-	-
S_{11}	8	5	1	1323	1296	56	393	375	21	-	-	-
S_{12}	13	7	1	791	763	55	12	1	2	82107	82047	6649

Table 3. Magic Squares: Enumerations (E), Backtracking (B), CPU time (t) in ms

		S_1			S_2			S_3			S_4		
Source	Degree	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)	(E)	(B)	(t)
[14]	None-1	84	52	14	220	195	21	1308	1283	88	183	159	26
[14]	None-2	2836	2815	153	271	249	23	11074	11048	603	124	102	22
[18]	Easy	7	3	11	17	13	11	7	3	10	17	13	12
[18]	Medium	16	6	11	174	164	19	16	6	11	174	164	26
[18]	Hard	27	16	11	24	18	11	27	16	11	24	18	12

Table 4. Sudoku solved with heuristic *MiD*

		S_5			S_6		
Source	Degree	(E)	(B)	(t)	(E)	(B)	(t)
[14]	None-1	-	-	-	-	-	-
[14]	None-2	-	-	-	-	-	-
[18]	Easy	18554	18537	1799	274476	274472	28149
[18]	Medium	-	-	-	121135	121113	12868
[18]	Hard	-	-	-	-	-	-

Table 5. Sudoku solved with S_5 and S_6 strategy

of the search space takes place because at the beginning of the process the variables chosen are related to a greater number of other variables (elements of the main diagonal), which produces that when evaluating the levels of consistency between the variables is eliminated a greater amount of inconsistency, reducing the space to explore.

In order to conclude, it is possible to mention that the size of the search space has a great incidence in the resolution process, where his exponential growth makes the process considerably more expensive, this is possible to appreciate when solving different instances of the same problem (of course, hypothesis demonstrated in a lot of previous work)

		S_7			S_8		
Source	Degree	(E)	(B)	(t)	(E)	(B)	(t)
[14]	None-1	-	-	-	-	-	-
[14]	None-2	-	-	-	-	-	-
[18]	Easy	24195	24169	2582	721773	72155	7484
[18]	Medium	-	-	-	88720	88706	9763
[18]	Hard	93138	93105	9158	-	-	-

Table 6. Sudoku solved with S_7 and S_8 strategy

With regard to the resolution of Sudoku, different published instances have been used from puzzles in the newspaper "The Times" [18], where usually the difficulty of puzzle is provided. On the other hand also a collection [14] of puzzles have been used which have single 17 numbers given in the initial matrix, this amount corresponds to the smaller known number given in the initial matrix [15], these last ones are not classified by difficulty.

In Table 4, Table 5 and Table 6 we show the source from where puzzles were obtained, the degree (difficulty level), the heuristic used to solve each instance and the different measured indicators of performance during the execution of the tests. Although some authors say that the amount of numbers given initially does not have incidence in the degree of difficulty of Sudoku, the results obtained here show bad results with 17 numbers in comparison with the other cases (easy, medium and hard), where the amount of numbers given initially is greater 25.

6 Conclusions

In this work we showed that variable and value selection heuristic influence the efficiency in the resolution of combinatorial problems. The efficiency of resolution was measured on the basis of performance indicators. The work included the modeling and resolution of classic puzzles (Magic Square, Latin Square and Sudoku) in Mozart.

The possibility to obtain better results in the search process was showed using suitable criteria of selection of variables and values. In fact, to select a variable in

a search process implies to determine the descending nodes of the present space that have a solution. It is very important to detect when the descending nodes are not a solution, because in this way we avoided to do unnecessary calculations that force to backtracking. Due to the above reasoning the heuristic selecting the variable with minimum domain size (*MiD*) presents a better behavior in comparison with the other strategies, because *MiD* bets by the variable going towards an insolvent space avoiding a priori unnecessary calculations.

We showed that the resolution possibilities of a certain problem depends on the search space size, it is possible to be appreciated when observing the differences generated in the results obtained in the resolution of different instances of the same problem.

At the moment of thinking in selection heuristics is important to consider the "model-and-run" paradigm for CP, in order to use variable and value selection heuristic efficiently.

In the future, we plan to integrate in a Constraint Programming process other techniques such as Local Search or Metaheuristics working like enumeration strategies.

References

1. K. Apt. Principles of constraint programming, 2003.
2. F. Barber and M. A. Salido. Introducción a la programación de restricciones. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 20:13–30, 2003.
3. R. Bartk. On-line guide to constraint programming. 1998. http://kti.mff.cuni.cz/ bartak/constraints/.
4. C. Beck, P. Prosser, and R. Wallace. Toward understanding variable ordering heuristics for constraint satisfaction problems. In *Fourteenth Irish Artificial Intelligence and Cognitive Science Conference - AICS 2003*, pages 11–16.
5. J. Borrett, E. Tsang, and N. Walsh. Adaptive constraint satisfaction: The quickest first principle. In W. Wahlster, editor, *ECAI*, pages 160–164. John Wiley and Sons, Chichester, 1996.
6. M. Cadoli, T. Mancini, and F. Patrizi. Sat as an effective solving technology for constraint problems. In F. Esposito, Z. Ras, D. Malerba, and G. Semeraro, editors, *ISMIS*, volume 4203 of *Lecture Notes in Computer Science*, pages 540–549. Springer, 2006.
7. C. Castro, E. Monfroy, C. Figueroa, and R. Meneses. An approach for dynamic split strategies in constraint solving. In A. F. Gelbukh, A. de Albornoz, and H. Terashima-Marín, editors, *MICAI*, volume 3789 of *Lecture Notes in Computer Science*, pages 162–174. Springer, 2005.
8. C. Castro, M. Moossen, and M.-C. Riff. A cooperative framework based on local search and constraint programming for solving discrete global optimisation. In A. Bazzan and S. Labidi, editors, *SBIA*, volume 3171 of *Lecture Notes in Computer Science*, pages 93–102. Springer, 2004.
9. R. Haralick and G. Elliot. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14:263 - 313, 1980.
10. T. Lambert, E. Monfroy, and F. Saubion. Solving sudoku with local search : A generic framework, May 28-31 2006. A paratre.

11. R. Lewis. Metaheuristics can solve sudoku puzzles. *In Press: Journal of heuristics*, 13, 2007.
12. E. Monfroy, C. Castro, and B. Crawford. Adaptive enumeration strategies and metabacktracks for constraint solving. In T. M. Yakhno and E. J. Neuhold, editors, *ADVIS*, volume 4243 of *Lecture Notes in Computer Science*, pages 354–363. Springer, 2006.
13. E. Monfroy, C. Castro, and B. Crawford. Using local search for guiding enumeration in constraint solving. In J. Euzenat and J. Domingue, editors, *AIMSA*, volume 4183 of *Lecture Notes in Computer Science*, pages 56–65. Springer, 2006.
14. G. Royle. Minimum sudoku. http://www.csse.uwa.edu.au/ gordon/sudoku17.
15. H. Simonis. Sudoku as a constraint problem. In B. Hnich, P. Prosser, and B. Smith, editors, *Proc. 4th Int. Works. Modelling and Reformulating Constraint Satisfaction Problems*, pages 13–27, 2005.
16. B. Smith. Succeed-first or Fail-first: A Case Study in Variable and Value Ordering. Technical Report 96.26, 1996.
17. B. M. Smith and P. Sturdy. Value ordering for finding all solutions. In L. P. Kaelbling and A. Saffiotti, editors, *IJCAI*, pages 311–316. Professional Book Center, 2005.
18. The Times. Spring su doku. April 2007.
19. E. Tsang. *Foundations of Constraint Satisfaction.* Academic Press, London, 1993.
20. T. Yato and T. Seta. Complexity and completeness of finding another solution and its application to puzzles. *IPSJ SIG Notes*, 2002(103):9–16, 20021108.
21. P. Zoeteweij. *Composing Constraint Solvers.* Printed and bound by PrintPartners Ipskamp, Enschede, 2005.

Comparison between Genetic Algorithms with Exaptation and Case Injected Genetic Algorithm in dynamic job shop problems

Luis Torres-Treviño

Corporación Mexicana de Investigación en Materiales
Oceania 190, Fracc. Saltillo 400
Phone (844)-411-32-00
Saltillo, Coahuila, Mexico.
ltorres@comimsa.com.mx

Abstract. It is propose new evolutionary algorithms with exaptive properties to tackle dynamic job shop problems. Exaptation is a new theory with two implicit procedures of retention and reuse of old solutions. The retention of a solution involves some kind of memory and the reuse of a solution implies the adaptation of the solution to the new problem. The first algorithm proposed applies seeding techniques to reuse a solution and the second algorithm proposed uses memory with seeding techniques to retain and reuse solutions respectively. Both algorithms are compared with a simple genetic algorithm with random initialization (RIGA) and the Case Injected Genetic Algorithm (CIGAR). The test include two selection mechanism, the tournament selection of size two and the selection proposed by Eshelman.

Track category: Evolutionary Scheduling and Routing

1 Introduction

In recent years the optimization of dynamic problems has become a growing field of research. The real-world problems are not static, they exist in a dynamic environment and it is necessary to modify the current solution when a change is detected. It is necessary evolutionary algorithms that do not re-start in every change; these algorithms must take advantage of the population information to obtain a valid solution in a short time. It is expected a similar solution when there are a minimum change of the problems and a dissimilar solution when there are an important change of the problem.

Some examples of dynamic problems are job shop problems where there are changes in due time, changes in the number of machines, processing times, etc., and these changes imply a re-schedule in the job shop. Learning in dynamic environments is a desirable quality for mobile robots. Navigation represents a simultaneous problem of path planning and movement to the goal along the path. Finally consider an company where in some periods there is high demand

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 249–257

Received 15/06/07
Accepted 31/08/07
Final version 30/09/07

of products and in another periods there is low demand. Usually it is necessary an optimization algorithm to manage and control sources efficiently with this dynamic demand. All of these dynamic problems can be modeled with variables, the optimization function and a set of constrains. Every one of them can changes through time [1]. Many authors have suggested some extension in the simple genetic algorithm to tackle these dynamic problems. Branke has suggested the following categories to group the algorithms proposed [2, 3]:

- Evolutionary algorithms that detect every change in the environment. If it is detected some change, then new individuals are injected into the population to increase diversity.
- Evolutionary algorithms which have an implicit memory. These algorithms use double or more complex representations (diploid, haploids) [4, 5]. In a given moment just one representation is active.
- Evolutionary algorithms which have an explicit memory to store useful information of the past and it is recalled when the dynamic problem returns to a similar situation presented in the past [6, 7].
- Evolutionary algorithms which avoid every time the convergence. Genetic algorithms with sharing and random immigrants are examples of these kinds of algorithms.
- Evolutionary algorithms that use a multiple subpopulations to search the optimum solution or to search a new one.

In the following sections it will be presented a comparison of two new evolutionary algorithms inspired in exaptation, versus the simple genetic algorithms with random initialization (RIGA), and the Case Injected Genetic Algorithm (CIGAR) proposed by Sushil J. Louis to solve dynamic job shop problems. Section II reviews the exaptation theory. In section III it is proposed two genetic algorithms inspired in exaptation theory. Section IV reports some experiments and results of the comparisons. Finally, the conclusions and future work will be give in the last section.

2 Exaptation

Gould and Vrba [8, 9] proposed the term exaptation, which refers to a trait that current provides fitness, but originally arose for some other reason. Every entity (species) tries to survive in a continuous non-static environment. The entity has traits which lets survive in the environment. Some traits are useful because provide high fitness but another ones do not provide fitness, they are usually useless. Some traits may have evolved in one context of the environment but later, such a trait may be co-opted for use in a different role. In other words, the exaptation is a change in the function of an old trait to solve a new problem similar or very different to the original one.

It is possible to describe three procedures in exaptation. First, when there is a change in the environment it is detected a set of possible traits with high fitness. Second, it is possible to reuse useful traits with high fitness and adapt

them to the new environment. The third procedure retains a useful trait for future references; however, the useless traits do not disappear completely, they are stored as redundant or useless structures.

The initial population of a genetic algorithm is random. When the simple genetic algorithm (SGA) solves a problem, it takes several cycles to get an optimum or an individual with high fitness. In the last generation the population has individuals that are very similar between them, it means, there are a low diversity. If there is a change of the environment (i.e. change the objective function) and some individuals are useless maybe some of them have useful traits. These traits are components of the phenotype so there are some genes to represent them. If the SGA runs again with a changed in the environment and the same population (without a random initialization), the useful traits may arise and they can let to get a new optimum quickly. In the SGA it is possible to reuse the last population if the function does not change too much. If the change is important then the reuse of the last population can be useless. It is necessary to modify the SGA in order to get some features of exaptation and it can be used to solve dynamic problems.

3 Genetic algorithms with exaptation and CIGAR

The first algorithm proposed is the SGA with seeding techniques; this algorithm is inspired in exaptation because it is reused some structures when the algorithm detects any change in the optimization function. If this change is detected then some variations or neighborhoods of the best solution found are injected into the actual population. The variations or neighborhoods replace a percent of the total population. It could be that some components (genes) of the best individual can be reused to improve the fitness in the new optimization function. The neighborhoods of the solution can give the appropriate solution if the problem changes slightly. The variations (mutations) of the solution can give a key to improve the fitness if the problem changes abruptly. This algorithm can be classified as an approach that detect a change and it is applied an injection.

The second algorithm is inspired in exaptation too and it is implemented from the point of view of learning by analogy; this learning mechanism has a memory of useful solutions of the past, a storing procedure of solutions, a search mechanism and a modification procedure. The second algorithm applies similar procedures: A recognition procedure where it is used the evaluation of the best individual to be compared again the best evaluation found in the past. If there is a degradation of the past solution then there is a change in the objective function. Storing procedure saves the best individual found into the memory. This avoids to store the same individual in the memory two times. The procedure is simple; first, it locates the most similar individual of the memory to the best individual found and if the best individual found has better evaluation than the individual of the memory then the best individual is stored into the memory. In other case, there is not change in the memory. This procedure tries to apply the exaptive

property of retention of solutions. The modification procedure is based in seeding techniques [10].

The algorithms implemented are the Randomly Injected Genetic Algorithm (RIGA), the Case Injected Genetic Algorithm (CIGAR), the SGA with seeding (SGAS) and the SGA with seeding and memory (SGASM). All the algorithms reuse the last population where it is initialized just at the beginning of the run. The RIGA is a simple genetic algorithm with random initialization and this is shown below. P is the population, F_E retains the evaluation value of every individual in the population. I is an auxiliary variable and it retains the best individual of the population per generation. The evaluation value of the Individual I is saved in e.

Simple Genetic Algorithms

1) $P \leftarrow$Random initialization
2) $F_E \leftarrow$ Evaluation(P)
3) Get the best individual I and its evaluation e from F_E
4) $P \leftarrow$ Selection(P, F_E)
5) $P \leftarrow$ Reproduction(P) (crossover and mutation)
6) If end condition is not satisfied, then go to step (2)
7) End

The CIGAR proposed by Sushil Louis consists of a case based reasoning system (CBRS) with a genetic algorithm. The components of each case are: the best solution, the number of generation in which it was stored, the value of evaluation, and the description of the problem to detect similarities. CBRS can place solutions of examples to have some reference in the search. The system initiates in random form and solutions are injected when some cases are similar to the new problem. In each certain number of generations, solutions of the CBRS are injected into the population of the genetic algorithm when the problem is similar to one already solved. The system looks for best the individuals of the population and it is stored in the memory of the CBRS, replacing the most similar solution but with smaller value of evaluation. The process is repeated in certain number of generations; for example, solutions in generations 1, 25, 50 and 100 can be stored and be injected. There are four forms in CIGAR to inject solutions, the first form consists of injecting solutions that are similar to the best-found solution; the second form consists of injecting solutions that are similar to the worse found solution. In the third form, solutions of CBRS are injected and chose according to a probability factor that is inversely proportional to the Hamming distance between the solution stored in the CBRS and the best solution found in the population. In the fourth form, solutions of the reasoning system are injected according to a probability factor that is directly proportional to the Hamming distance between the cases of the reasoning system and a solution of the population. In all the cases the injected solutions replace the worse individuals of the population. Lately the author has used his system CIGAR without the mechanism to detect similarities because it implies a high cost or a difficulty to determine a metric for certain problem. In was replaced by the fitness evaluation [12–14].

The CIGAR algorithm is shown below. In these experiments was used the fitness value as a metric to detect similarities and it is injected the solutions of the memory I_M (extracted) that are similar to the best-found solution of the population [16–18]

Cased Injected Genetic Algorithm
1) $P \leftarrow$ Random initialization
2) $M \leftarrow$ Random initialization or empty memory
3) $F_E \leftarrow$ Evaluation(P)
4) Get the best individual I and its evaluation e from F_E
5) $P \leftarrow$ Selection(P, F_E)
6) $P \leftarrow$ Reproduction(P) (crossover and mutation)
7) Every 25 generations save the best solution I into the memory M (case based)
8) Every 25 generations extract the solution I_M (see text above) from M and inject it in P
9) If end condition is not satisfied, then go to step (3)
10) End

The genetic algorithm with seeding injects neighbors of the best individual I when it detects a change in the objective function.

Genetic algorithm with seeding techniques
1) $P \leftarrow$ Random initialization
2) Get the best past evaluation e_a
3) $FE \leftarrow$ Evaluation(P)
4) Get the best individual I and its evaluation value e from F_E
5) If $e < e_a$ then apply seeding of 50% of neighbor and variations of I in P
6) $P \leftarrow$ Selection (P, F_E)
7) $P \leftarrow$ Reproduction (P) (crossover and mutation)
8) Includes Elitism injecting I in P
9) If end condition is not satisfied, then go to step (2)
10) End

The SGA with memory and seeding has a memory to save the best solution found. The algorithm uses a SGA with seeding of neighborhoods and variations of 50% (25% each one). The rest is random. The condition to make a seeding is when the evaluation value of the memory e_m is greater than the best past evaluation value e_a. When the memory detects a small evaluation in the individuals of the memory this is an indication that the function changes and there is an unknown solution, so it is necessary to find a new one considering only the information of the memory [11].

Genetic algorithm with memory and seeding techniques

1) $P \leftarrow$ Random initialization
2) $M \leftarrow$ Random initialization or empty memory
3) Get the best past evaluation value e_a
4) $FEM \leftarrow$ Evaluation(M)
5) Get the best individual I_m and its evaluation value e_m from F_{EM}
6) If $em > e_a$ then applies seeding of 50% of Neighborn and variations of I_m in P
7) $FE \leftarrow$ Evaluation(P)
8) Get the best individual I and its evaluation value e from F_E
9) $P \leftarrow$ Selection(P, F_E)
10) $P \leftarrow$ Reproduction(P) (crossover and mutation)
11) Every k generations saves in memory M the best individual I
12) If end condition is not satisfied, then go to step (3)
13) End

It is important to describe the injection mechanism of the genetic algorithms with exaptation. The procedure requires the injection of variations and neighborhoods that represent sequence solution. The generation of neighborhoods use a method called one step look-back interchange [19]. It is necessary a sequence of operations and the list of machines where the operations were allocated. The method first locate a job k that was assigned in machine m (called position p_1) and locate another operation which has the same machine m (called position p_2). In the next step it is interchanged the operations between position $p1$ and $p2$ so we have a new sequence. The solutions variation is a simpler procedure where it is selecting two random positions of operations and it is interchanged the operations among them.

4 Experimentation and results

This experiment is inspired in Sushil Louis works [15]. The problem is to minimize the makespan; that means, the total elapsed time between the beginning of the first task and the completion of the last task. In the problem of dynamic JSP there are 10 jobs and 10 machines; each job comprises a set of operations or task. The operations must each be done in different machines in a given job-dependent order.

A machine does not process two different jobs at once and different operations of the same job are not simultaneously being processed on different machines. The sequence of operations or tasks represents a chromosome of size 100 (10 jobs x 10 operations per job). The enconding follow the scheme proposed by Fang [20].

It is generated with a uniform random initialization 10 jobs with 10 operations. Each job has a due time between 20 and 500; the processing time of every

job is between 4 and 12 time units and the job-dependent order has a uniform random initialization. This is called the basic problem.

It is generated 50 problems picking randomly the processing time of the operations (40% of the total operations) from the basic problem and the processing time has a random modification adding or resting up to 4 units of time.

For the experiment all the evolutionary algorithms has a crossover probability of 0.9 and mutation rate of 0.0725, it is used a swap mutation and a greedy crossover [21], a total population size of 100 individuals. The CIGAR memory has the capacity to save 50 solutions (cases) and the SGASM memory has the capacity to store 10 solutions. It is used a selection tournament of size 2. Every generation involves 100 evaluations per generation. All the algorithms solve the same problems. It is reported the best solution found per algorithm per problem.

It is used two selection procedures. The selection procedure proposed by Eshelman [22] was used by Sushil Louis in his CIGAR system. Every plot illustrates the performance to minimize the total cost of makespan in every problem through time. The figure 1 shows all the algorithms for the dynamic JSP problem with selection proposed by Eshelman. The plot suggests a better performance of the genetic algorithms with seeding and memory.

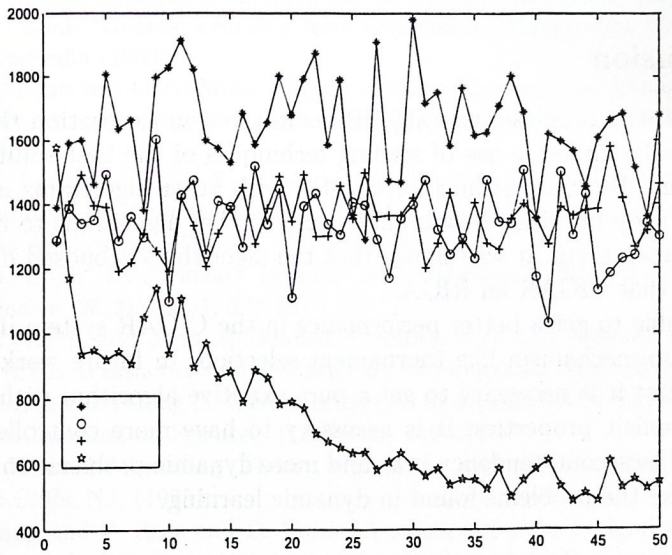


Fig. 1. Performance of several approaches in every problem using selection proposed by Eshelman.

It was changed the selection procedure and it is used the tournament selection of size two in all algorithms. The figure 2 shows a better performance of the genetic algorithms that use exaptation.

RIGA has a poor performance because do not use the last information and take the every problem as a new one.

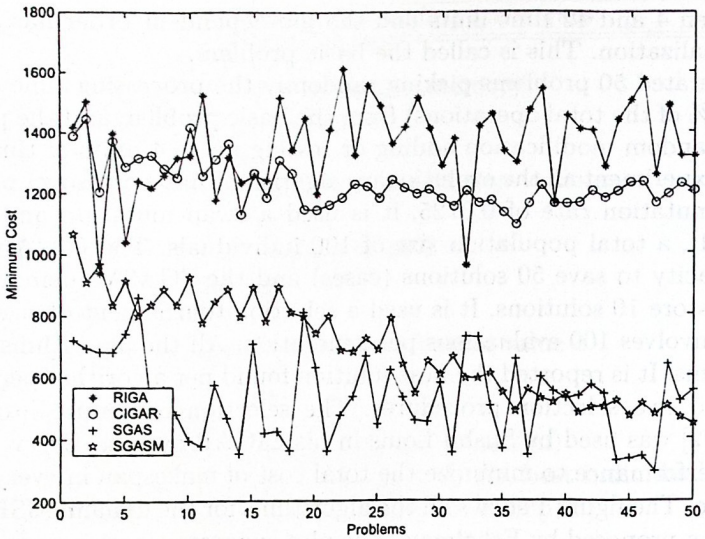


Fig. 2. Performance of several approaches in every problem using tournament selection of size 2.

5 Conclusion

In this paper it is proposed two algorithms inspired in exaptation theory. The algorithms make intensive use of seeding techniques of the best solution useful at the moment. Both algorithms were tested with other algorithms in a simple dynamic job shop problem. It was implemented other algorithms to make comparisons between them. It was shown that the algorithms proposed have better performance that CIGAR an RIGA.

It is possible to get a better performance in the CIGAR system if it is used other selection mechanism like tournament selection. In future work there are two routes, first it is necessary to get a pure exaptive algorithm with retention and reuse implicit properties. It is necessary to have more controlled seeding techniques. The second tendency is to find more dynamic problems to test these algorithms like the problems found in dynamic learning.

References

1. Trojanowski, K. and Michalewicz, Z. Evolutionary algorithms for non-stationary environments. In: Proc. of 8th Workshop: Intelligent Information systems, ICS PAS Press, (1999) 229–240
2. Branke, J. Evolutionary approaches to dynamic optimization problems; a survey. In: GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, (1999) 134–37
3. Branke, J. Evolutionary approaches to dynamic optimization problems - Updated survey. In: GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems (2001), 27–30

4. Goldberg, D. Genetic algorithms in search, optimization, and machine learning. Addinson - Wesley. Pub. Co., Reading MA (1989).
5. Dasgupta, D. and McGregor, D. Nonstationary function optimization using the structured genetic algorithm. In: R. Männer and B. Manderick, editors, Parallel Problem Solving from Nature, Elsevier Science Publisher (1992) 145–154
6. Louis, Sushil and Johnson, J.: Robustness of Case-Initialized Genetic Algorithms, FLAIRS-99, Orlando, FL, AAAI Press, (1999)
7. Connie Loggia Ramsey and John J. Grefenstette: Case-Based Initialization of Genetic Algorithms. Proc. of the Fifth Int. Conf. on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA. (1993) 84–91.
8. S. J. Gould and S. Vrba: Exaptation: A missing term in the science of form, Paleobiology, volume 8 (1982) 4–15
9. Stephen Jay Gould: A crucial Tool for an Evolutionary Psychology, Journal of Social Issues, volume 47, (1991) 43–65
10. Torres-Treviño, L. Sistemas exaptivos: retención y reutilización de conocimiento en algoritmos evolutivos. Tesis Doctoral. Instituto Tecnológico de Estudios Superiores de Monterrey. Campus Monterrey (2004)
11. Torres, L. GA with Exaptation: New Algorithms to Tackle Dynamic Problems. Lecture Notes in Computer Science, 2972, (2004) 746-753
12. Sushil J. Louis and Judy Johnson: Solving Similar Problems Using Genetic Algorithms and Case-Based Memory, Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kauffman, San Mateo, CA, (1997) 283–290
13. Sushil J. Louis: Genetic Learning from Experience, Accepted in CEC-03, Camberra, Australia (2003)
14. Sushil J. Louis and Chris Miles. Playing to learn: Case-injected genetic algorithms for learning to play compuer games. IEEE Transactions on Evolutionary Computation, 9(6). 2005
15. Sushil J. Louis and Zhijie Xu: Genetic algorithms for Open Shop Scheduling and Re-Scheduling, M. E. Cohen and D. L. Hudson, ISCA 11th Int. Conf. on Computers and their Applications, Raleigh, NC (1996) 99–102
16. Sushil J. Louis. Evolutionary learning from experience. Journal of Engineering Optimization, 26(2), (2004) 237-247
17. Sushil J. Louis and John McDonnell. Learning with case injected genetic algorithms. IEEE Transactions on Evolutionary Computation, 8(4), (2004) 316-328
18. Sushil J. Louis and Gong Li. Case injected genetic algorithms for traveling salesman problems. Information Sciences, 122, (2000) 210-225.
19. Pinedo, Michael Scheduling: Theory, algorithms, and systems, Prentice Hall, Englewood Cliffs, NJ, (1995)
20. H. L. Fang and P. Ross and D. Corne: A promising genetic algorithm approach to job shop scheduling, rescheduling, and open shop scheduling problems, Proc. of the Fifth Int. Conf. on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA., (1993) 375–382
21. G. E. Liepins and M. R. Hilliard and M. Palmer and M. Morrow: Greedy genetics, In Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum, (1987) 90–99
22. L. J. Eshelman: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, Foundations of genetic algorithms 1, (1991) 265–283.

Research on Dynamic OD Analysis for Urban Traffic

DONG Hong-zhao[1,2], ZHOU Min[1,2], GUO Ming-fei[1,2],Qian Xiao-hong[2]

[1]MOE Key Laboratory of Mechanical Manufacture and Automation,
Zhejiang University of Technology, Hangzhou 310014, Zhejiang Province, China
[2] ZJUT-ENJOYOR ITS Joint Institute,Zhejiang University of Technology,
Hangzhou 310014, Zhejiang Province, China
zhouminchinahz@gmail.com

Abstract. Traditional dynamic OD (Origin-destination) analysis is an important method of acquiring traffic information in urban traffic studies. However, just one single parameter of the traffic flow could be obtained from the traffic flow database, and it is always fractional and uncertain. A new dynamic OD analysis method based on time slice is put forward here to obtain traffic information more comprehensive than before. First, the length of time slice, a necessary factor to preprocess the dynamic OD data could be estimated in terms of historical OD matrix. Consequently, the OD data can be analyzed to get traffic information within each time slice. More traffic parameters within an interval of time such as travel time, vehicle volume etc., can be obtained by combining the results of each time slice. Finally, a practical dynamic OD analysis system is developed. The results showed the comprehensive traffic parameters can be obtained resorting to the proposed method.

1 Introduction

In recent years, the research and application of intelligent transportation systems (ITS) have spread all over the world. The dynamic self-adapting traffic management is the goal of ITS. To attain the object, time-based OD matrix is used to reflect traffic demand with various intervals (such as per hour, per minute, etc.).

The OD matrix can reflect the traffic status in the road grid effectively. The OD analysis becomes a mainstream method to obtain traffic parameters. Generally, to obtain OD matrix, there are two main methods including direct and indirect methods as shown in Fig1. [1][2]

The dynamic OD matrix has the features of time variance and uncertainty. The previous research on OD matrix is mainly focused on static OD matrix., in which traffic information between any OD pairs can be obtained by static multi-route assignment model [3].

Nowadays, the research is confined in the estimation and prediction of the dynamic OD matrix, which is mostly applied in the simple road grid such as highways and belt

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 258–268

Received 17/06/07
Accepted 31/08/07
Final version 21/09/07

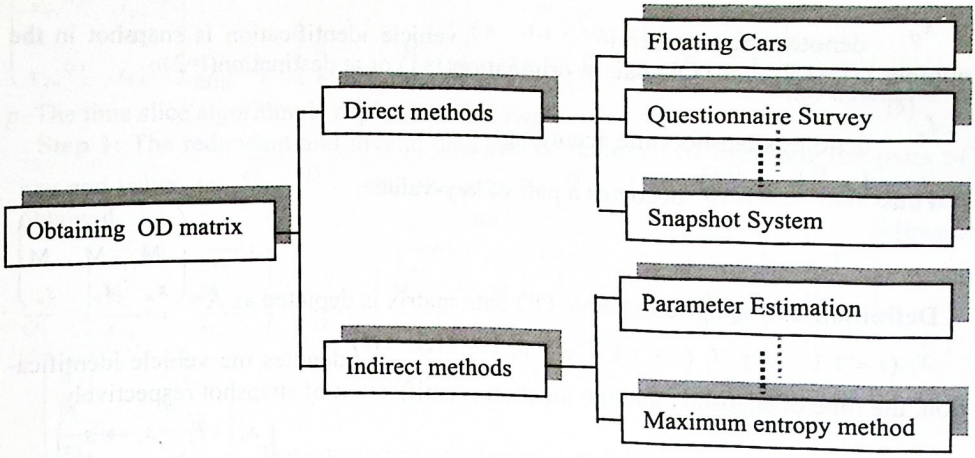


Fig. 1. The methods of getting OD matrix

highways [4].The methods such as the least square method [5] and the predication of dynamic route travel time based on fuzzy judgment [6], etc. can not suit to the complex urban road grid.

A new dynamic OD analysis method is put forward here. An interval is further divided into short time slices. In each time slice, with the calculation of vehicle data including its location and time coming from the snapshot, the traffic parameters within an interval of time can be obtained by combining each time slice. The method can effectively obtain more traffic parameters in traffic grid such as route travel time, vehicle volume, and so on.

2 Establishment of Model

Because there is the uncertainty in the dynamic OD matrix, a new strategy is proposed that the whole period is divided into small time slices. In each slice, the dynamic OD matrix is approximately considered as a static OD matrix, which is processed to acquire some parameters. The actual traffic information of whole period can be imitated through integrating the parameters in all time slices.

2.1 Definition of Concepts

Based on the strategy above, the following concepts are defined:

Definition 1: The set of OD pairs is depicted as $\{(O_k \quad D_k) \mid k = 1,2,\mathrm{L} \quad ,m\}$, and the snapshot records are $O_k = \begin{pmatrix} x_{11}^{(k)} & t_{11}^{(k)} \\ \mathrm{M} & \mathrm{M} \\ x_{1n}^{(k)} & t_{1n}^{(k)} \end{pmatrix}$, $D_k = \begin{pmatrix} x_{21}^{(k)} & t_{21}^{(k)} \\ \mathrm{M} & \mathrm{M} \\ x_{2n}^{(k)} & t_{2n}^{(k)} \end{pmatrix}$.

$x_{ij}^{(k)}$ denotes the number $j(j=1,L\ ,n)$ vehicle identification is snapshot in the number k(k=1,2,…,m) OD pair at origination(i=1) or at destination(i=2).

$t_{ij}^{(k)}$ denotes snapshot time accordingly.

Thus $\left(x_{ij}^{(k)} \quad t_{ij}^{(k)}\right)$ becomes a pair of key-value.

Definition 2: Road grid dynamic OD data matrix is depicted as $A=\begin{pmatrix} x_1 & t_1 & s_1 \\ M & M & M \\ x_n & t_n & s_n \end{pmatrix}$.

$x_i\ (i=1,L\ ,n)$, $t_i\ (i=1,L\ ,n)$, $s_i\ (i=1,L\ ,n)$ denotes the vehicle identification, the time of snapshot, and the location identification of snapshot respectively.

Definition 3: The road travel time matrix is depicted as $C=\begin{pmatrix} x_1 & \overline{T_1} & s_i \to s_j \\ M & M & M \\ x_n & \overline{T_n} & s_k \to s_n \end{pmatrix}$,

$x_i\ (i=1,L\ ,n)$ denotes the vehicle identification.

$\overline{T_i}\ (i=1,L\ ,n)$ denotes the time interval of the two snapshot time. $s_i \to s_j\ (i=1,L\ ,n; j=1,L\ ,n)$ denotes the route from s_i to s_j that the same vehicle passes.

2.2 Selection of Time Slice

The time slice can be acquired by analyzing the history data of every cross-road. Because the traffic flow has a periodicity feature, the target OD pair's time slice can be obtained by matching with history OD matrix under appropriate threshold of vehicle volume [8]. The selection of threshold is very important. If the threshold becomes larger, the precision will be decreased. If it becomes smaller, then process time spending will increase, even induces the "Runger phenomenon"— the smaller the time slice is, the larger the error becomes.

2.2.1 Time slice algorithm of two OD pairs

Considering two OD pairs---$(O_1 \quad D_1)$, $(O_2 \quad D_2)$, their history matrixes of snapshot records are O_1 $\begin{pmatrix} x_{11}^{(1)} & t_{11}^{(1)} \\ M & M \\ x_{1n}^{(1)} & t_{1n}^{(1)} \end{pmatrix}$ D_1

$\begin{pmatrix} x_{21}^{(1)} & t_{21}^{(1)} \\ M & M \\ x_{2n}^{(1)} & t_{2n}^{(1)} \end{pmatrix}$ and O_2 $\begin{pmatrix} x_{11}^{(2)} & t_{11}^{(2)} \\ M & M \\ x_{1n}^{(2)} & t_{1n}^{(2)} \end{pmatrix}$ D_2 $\begin{pmatrix} x_{21}^{(2)} & t_{21}^{(2)} \\ M & M \\ x_{2n}^{(2)} & t_{2n}^{(2)} \end{pmatrix}$, the threshold is p. The time slice algorithm is depicted as follows:

Step 1: The redundant and invalid data can be removed by matching the pairs of key and value in O_1 D_1 with the ones in O_2 D_2, then new matrixes are obtained as follows:

$\overline{O_1}$ $\begin{pmatrix} \overline{x_{11}}^{(1)} & \overline{t_{11}}^{(1)} \\ M & M \\ \overline{x_{1m}}^{(1)} & \overline{t_{1m}}^{(1)} \end{pmatrix}$ $\overline{D_1}$ $\begin{pmatrix} \overline{x_{21}}^{(1)} & \overline{t_{21}}^{(1)} \\ M & M \\ \overline{x_{2m}}^{(1)} & \overline{t_{2m}}^{(1)} \end{pmatrix}$ and $\overline{O_2}$ $\begin{pmatrix} \overline{x_{11}}^{(2)} & \overline{t_{11}}^{(2)} \\ M & M \\ \overline{x_{1h}}^{(2)} & \overline{t_{1h}}^{(2)} \end{pmatrix}$ $\overline{D_2}$ $\begin{pmatrix} \overline{x_{21}}^{(2)} & \overline{t_{21}}^{(2)} \\ M & M \\ \overline{x_{2h}}^{(2)} & \overline{t_{2h}}^{(2)} \end{pmatrix}$. The constraints are depicted as follows:

$m \le n, h \le n$

$$\overline{x_{1i}}^{(1)} = \overline{x_{2i}}^{(1)} \in \{x_{1i}^{(1)} \mid i = 1,...m\} \cap \{x_{2i}^{(1)} \mid i = 1,...m\} \qquad \overline{t_{1i}}^{(1)} < \overline{t_{2i}}^{(1)}$$

$$\overline{x_{1i}}^{(2)} = \overline{x_{2i}}^{(2)} \in \{x_{1i}^{(2)} \mid i = 1,...h\} \cap \{x_{2i}^{(2)} \mid i = 1,...h\}$$

$$\overline{t_{1i}}^{(2)} < \overline{t_{2i}}^{(2)} \quad ;$$

Step 2: Assuming the threshold is p, $m \le h$, there are three conditions below:

i. if $p>h$, then the OD pair O_1 D_1 and O_2 D_2 don't satisfy the threshold condition, the algorithm stop.

ii. If $p \in (m, h]$, then only O_2 D_2 satisfies the condition. So p corresponding records can be chosen randomly from $\overline{O_2}$ $\overline{D_2}$. The result is depicted as follows:

$\overline{\overline{O_2}}$ $\begin{pmatrix} \overline{\overline{x_{11}}}^{(2)} & \overline{\overline{t_{11}}}^{(2)} \\ M & M \\ \overline{\overline{x_{1p}}}^{(2)} & \overline{\overline{t_{1p}}}^{(2)} \end{pmatrix}$ $\overline{\overline{D_2}}$ $\begin{pmatrix} \overline{\overline{x_{21}}}^{(2)} & \overline{\overline{t_{21}}}^{(2)} \\ M & M \\ \overline{\overline{x_{2p}}}^{(2)} & \overline{\overline{t_{2p}}}^{(2)} \end{pmatrix}$;

iii. If $p \le m$, then the two pairs both satisfy the condition. So p corresponding records can be chosen randomly from $\overline{O_1}$ $\overline{D_1}$ $\overline{O_2}$ $\overline{D_2}$. The result is depicted as follows:

$\overline{\overline{O_1}}$ $\begin{pmatrix} \overline{\overline{x_{11}}}^{(1)} & \overline{\overline{t_{11}}}^{(1)} \\ M & M \\ \overline{\overline{x_{1p}}}^{(1)} & \overline{\overline{t_{1p}}}^{(1)} \end{pmatrix}$ $\overline{\overline{D_1}}$ $\begin{pmatrix} \overline{\overline{x_{21}}}^{(1)} & \overline{\overline{t_{21}}}^{(1)} \\ M & M \\ \overline{\overline{x_{2p}}}^{(1)} & \overline{\overline{t_{2p}}}^{(1)} \end{pmatrix}$; $\overline{\overline{O_2}}$

$$\begin{pmatrix} \overline{\overline{x}}_{11}^{(2)} & \overline{\overline{t}}_{11}^{(2)} \\ \mathrm{M} & \mathrm{M} \\ \overline{\overline{x}}_{1p}^{(2)} & \overline{\overline{t}}_{1p}^{(2)} \end{pmatrix} \quad \overline{\overline{D}}_2 \quad \begin{pmatrix} \overline{\overline{x}}_{21}^{(2)} & \overline{\overline{t}}_{21}^{(2)} \\ \mathrm{M} & \mathrm{M} \\ \overline{\overline{x}}_{2p}^{(2)} & \overline{\overline{t}}_{2p}^{(2)} \end{pmatrix}$$

Step 3: If $p \in (m, h]$, the time interval between the earliest time of $\overline{\overline{O}}_2$ and the latest time of $\overline{\overline{D}}_2$ can be obtained. For example, the earliest time of $\overline{\overline{O}}_2$ is $t_{o_2} = \min(\overline{\overline{t}}_{1i}^{(2)})$, and the latest time of $\overline{\overline{D}}_2$ is $t_{D_2} = \max(\overline{\overline{t}}_{2i}^{(2)})$ i 1,2... p , the time interval is t t_{D_2} t_{o_2}. So the time slice is Δt t; If $p \leq m$, then the two time interval between $\overline{\overline{O}}_1$ and $\overline{\overline{D}}_1$, $\overline{\overline{O}}_2$ and $\overline{\overline{D}}_2$ can be obtained. For example, the earliest time of $\overline{\overline{O}}_1$ is $t_{o_1} = \min(\overline{\overline{t}}_{1i}^{(1)})$, the latest time of $\overline{\overline{D}}_1$ is $t_{D_1} = \max(\overline{\overline{t}}_{2i}^{(1)})$;and the earliest time of $\overline{\overline{O}}_2$ is $t_{o_2} = \min(\overline{\overline{t}}_{1i}^{(2)})$, the latest time of $\overline{\overline{D}}_2$ is $t_{D_2} = \max(\overline{\overline{t}}_{2i}^{(2)})$ i 1,2... p , the two time intervals are respectively t_1 t_{D_1} t_{o_1} and t_2 t_{D_2} t_{o_2}. To satisfy the threshold condition, Δt max t_1 t_2 is selected as the time slice.

2.2.2 Time slice algorithm of multiple OD pairs

If the set of OD pairs is depicted as $\{(O_k, D_k) \mid k = 1,2,\mathrm{L}\ ,m\}(m \geq 3)$, then the algorithm can be shown as follows:

Step 1: Because of the space similarity of traffic volume, the two neighboring OD pairs can be treated as a group: $\{(OD_k, OD_{k+1}) \mid k = 1,3,\mathrm{L}\ ,m-1\}$ OD_k denotes (O_k, D_k)

Step 2: Call the algorithm depicted in section 2.2.1 to figure out the time slice of each group. The result is $\{\Delta t_k \mid k=1\ 3\ \mathrm{L}\ ,m-1\}$;

Step3: The time slice of the area road grid can be obtained as follows: $\Delta t = \sup\{\Delta t_k \mid k=1\ 3\ \mathrm{L}\ ,m-1\}$.

2.3 Algorithm of Travel Time

The travel time refers to the average time that the vehicles spend when passing the neighboring road crosses. The dynamic OD matrix constructed from snapshot data is divided by the time slice. The dynamic OD matrix is analyzed with static OD analysis

during each time slice. The algorithm can be shown as follows:

Step 1: The dynamic OD matrix data are filtered in each time slice Δt according to the initial time t in order to obtain the domain that the algorithm runs. For example, the OD matrix data transmitted from optical receiver is depicted as A= $\begin{pmatrix} x_1 & t_1 & s_1 \\ M & M & M \\ x_n & t_n & s_n \end{pmatrix}$ .then the filtrated data is $\overline{R}$. $\overline{R}$ $\begin{pmatrix} x_1 & t_1 & s_1 \\ M & M & M \\ x_m & t_m & s_m \end{pmatrix} (m \le n, t \le t_i \le t + \Delta t)$

Step 2: Because there exists redundancy in the matrix data, the data must be preprocessed with the vehicle identification matching to remove the abnormal or single data. The algorithm is depicted as follows:

i. x x_1

ii. Traverse the record matrix $\overline{R}$ with x

iii. If $\exists x_i = x$, add the x and x_i corresponding records into the set R , and at the same time delete those corresponding records in $\overline{R}$. If there is only one record left in $\overline{R}$,go to the step ; otherwise rearrange the records and give value x_1 to the first record, then go to step i;

iv. If $\forall x_i \neq x$, delete the record that the x corresponds to in $\overline{R}$. If there is only one record in the $\overline{R}$,go to the step ;else rearrange the records and give value x_1 to the first record, then go to step i;

v. The algorithm ends;

The final result is R: R= $\begin{pmatrix} x_1 & t_1 & s_1 \\ M & M & M \\ x_h & t_h & s_h \end{pmatrix}$ ($h \le m$, the vehicle identification $x_i \quad i \in [1, n]$ corresponds to multiple records)

Step 3: Compute the travel time from the OD matrix data obtained from step 2. If the count of records that x_i corresponds to is *k*, the *k* records are depicted as R_i : $R_i = \begin{pmatrix} u & t_i & s_i \\ M & M & M \\ u & t_{i+k-1} & s_{i+k-1} \end{pmatrix}$. Then do subtraction of the time field of every two re-

cords in R_i . C_k^2 records can be got. The result is depicted as $C_i : C_i \begin{pmatrix} u & T_i & s_i \to s_i \\ M & M & M \\ u & T_{i+C_i^2-1} & s_{i+k-1} \to s_{i+k-1} \end{pmatrix}$

Step 4: Filtrate C_i with the use of adjacent relationship of two road-crosses in road grid topological structure. The result is $\overline{C_i} : \overline{C_i} \begin{pmatrix} u & \overline{T_1} & s_i \to s_j \\ M & M & M \\ u & \overline{T_m} & s_k \to s_n \end{pmatrix}$ ($\overline{T_i} > 0$ $0<i<m$ s_i adjacent to s_j s_k adjacent to s_n)

Step 5: Loop step 3 and 4. The travel time matrix $\overline{C_i}$ $1 \le i \le h$ of every group data are obtained. The travel time matrix between two adjacent road-crosses of one vehicle can be depicted as partitioned matrix $\overline{C} : \overline{C} = \left(\overline{C_1} \quad \overline{C_2} \text{ L} \quad \overline{C_{h-1}} \quad \overline{C_h}\right)^T$

Step 6: The records in $\overline{C}$ are the travel time of single vehicle, the road average travel time is the weighted mean of the records in $\overline{C}$. For example, the records the road $s_i \to s_j$ corresponds to in $\overline{C}$ are $D_i : D_i \begin{pmatrix} x_m & \overline{T_m} & s_i \to s_j \\ M & M & M \\ x_n & \overline{T_n} & s_i \to s_j \end{pmatrix}$. Vehicle i is assigned a weight α_i according to the vehicle type x_i . The weight α_i can be determined by many methods such as statistical mean value method binomial coefficient method paired comparison method CCM analytic hierarchy process etc. Now the matrix is depicted as $D_i : D_i \begin{pmatrix} x_m & \overline{T_m} & s_i \to s_j \alpha_m \\ M & M & M \quad M \\ x_n & \overline{T_n} & s_i \to s_j \alpha_n \end{pmatrix}$. The road average travel time of $s_i \to s_j$ is $\overline{\overline{T_{ij}}} : \overline{\overline{T_{ij}}} = \sum_{k=m}^{n} \overline{T}_k \alpha_k$. The matrix D_i only includes the information of road and travel average time. The final result C is combined from D_i : $C = \left(D_1 \quad D_2 \quad \text{L} \quad D_h\right)^T$

3 Application

This research has been applied in the intelligent traffic guidance information platform of Hangzhou city China. The system's main functions are include the inquiry of traffic flow, Graphic inquiry and Exportation of excel.

3.1 Running Procedure of the System

3.1.1 Computation of Time Slice

In this part, two road segments are taken for example which are from Yan'an-Qingcun roadcross to Qingcun-Zhonghe roadcross and from Linying-Shuguang roadcross to Huanglong-Shuguang roadcross respectively.The OD matrix data are sampled from the four roadcrosses above. The time segment is from 10:10:00 a.m., Jan 3th, 2006 to 10:20:00 a.m., Jan 3th, 2006. The threshold is 10 vehicles. 10 records are selected from those samples randomly. The matching results are shown in Table 1 and Table 2:

From the two tables above, the earliest time of 10 vehicles passed Yan'an-Qincun roadcross is t_{o_1} =10:12:11, the latest time of Qincun-Zhonghe roadcross is t_{D_1} =10:16:24. Then the time interval is $t_1 \quad t_{D_1} \quad t_{o_1}$ =4 minutes and 13 seconds.

Similarity, the earliest time of 10 vehicles passed Linying-Shuguang roadcross is t_{o_1} =10:10:48, the latest time of Huanglong-Shuguang roadcross is t_{D_1} =10:15:48. The time interval is $t_1 \quad t_{D_1} \quad t_{o_1}$ =5 minutes.

According to the algorithm above, time slice is Δt: $\Delta t \quad \max \quad t_1 \quad t_2$ =5 min.

3.1.2 Computation of Travel Time

With the time slice Δt, we can obtain the travel time from Yan'an-Qingcun roadcross to Qingcun-Zhonghe roadcross in the time between 10:10:00 a.m., Jan 10th, 2006 and 10:20:00 a.m., Jan.10th, 2006.

First, the data are further filtrated according to the time slice and system trigger time at which the system starts to run. For example, if the system trigger time is 10:17:00 a.m. Jan. 10th,2006, then the processing period is from 10:12:00 a.m. Jan 10th,2006 to 10:17:00 a.m. Jan. 10th,2006. Second, every two of the filtered data make subtraction with each other. Then the mean value can be acquired (the vehicle type of the vehicles in this sample are considered the same). Table 3 shows process.

The time intervals of each sample are as follows: $\overline{T_1}$ =1min 40sec $\overline{T_2}$ =2 min 57 sec $\overline{T_3}$ =2 min 44 sec $\overline{T_4}$ =3 min 20 sec $\overline{T_5}$ =2 min 29 sec $\overline{T_6}$ =2min 52sec $\overline{T_7}$ =3min4sec $\overline{T_8}$ =2min41sec $\overline{T_9}$ =2 min 52 sec $\overline{T_{10}}$ =2 min 06 sec $\overline{T_{11}}$ =2 min 38 sec. The weight is α_k: α_k =1/11 k=1 2 L 11 . And finally

the average travel time is depicted as $\overline{\overline{T}}$: $\overline{\overline{T}} = \sum_{i=1}^{11} \overline{T}_i \alpha_i$ =2 min 40 sec with the algorithm above.

Table 1. The 10 samples randomly obtained from Yan'an-Qingcun No.1 and Qingcun-Zhonghe(No2) roadcrosses

Samples	1	2	3	4	5	6	7	8	9	10	Extreme early/late time	Interval
Vehicle ID	A36BXX	A3T6XX	A56JXX	A56UXX	B7E4XX	AT79XX	A70XXX	AT78XX	AT5XX	B33IXX		
No.1	10:12:37	10:13:20	10:14:20	10:13:12	10:13:22	10:12:11	10:12:30	10:14:20	10:12:54	10:12:34	10:12:11	4min 13sec
No.2	10:14:39	10:16:07	10:16:20	10:16:03	10:16:24	10:15:55	10:15:55	10:16:10	10:14:59	10:15:51	10:16:24	

Table 2. The 10 samples randomly obtained from Linyin-Shuguang(No.1) and Huanglong-Shuguang(No.2) roadcrosses

Samples	1	2	3	4	5	6	7	8	9	10	Extreme early/ late time	Interval
Vehicle ID	A456XX	AT36XX	A54YXX	B674XX	B454XX	AT56XX	A77IXX	AT9P3XX	A165XX	A6HJXX		
No.1	10:11:26	10:11:20	10:14:20	10:10:48	10:11:20	10:10:37	10:11:20	10:11:15	10:13:20	10:14:20	10:10:48	5 min
No.2	10:13:37	10:14:13	10:15:42	10:13:37	10:14:55	10:13:18	10:14:20	10:14:11	10:14:10	10:15:48	10:15:48	

Table 3. The computation of travel time

Samples	1	2	3	4	5	6	7	8	9	10	11	Time mean value
Vehicle ID	AT28XX	A471XX	A13XX	A53XX	A57QXX	AT63XX	B23XX	B67XX	A546XX	B465XX	AT10XX	
No.1	10:14:35	10:12:24	10:13:55	10:13:11	10:14:11	10:12:45	10:12:12	10:14:42	10:13:56	10:14:20	10:13:20	
No.2	10:16:15	10:15:21	10:16:39	10:16:31	10:16:40	10:15:37	10:15:26	10:16:23	10:16:48	10:16:26	10:15:58	
Time interval	1min40sec	2min57sec	2min44sec	3min20sec	2min29sec	2min52sec	3min4sec	2min41sec	2min52sec	2min06sec	2min38sec	2min40sec

3.2 Exhibition of Results

The graphic inquiry is a distinguishing function of the system, which is shown in Fig 2(in Chinese). In the query interface, the names of roadcrosses that we concern are input by clicking the location on the map. They are displayed in the two text boxes on the first line. The inquiry period between the input start time and the end time are displayed in the two text boxes on the second line. After the requisite information is all input, the road travel time will be shown as in Fig. 3(in Chinese).
The dynamic OD analysis is a basic and effective method of acquiring traffic parameters of road grid. To reduce the complexity and inaccuracy in traditional OD analysis, an improved dynamic OD analysis is put forward in this article which provides an effective approach to obtain the traffic parameters.

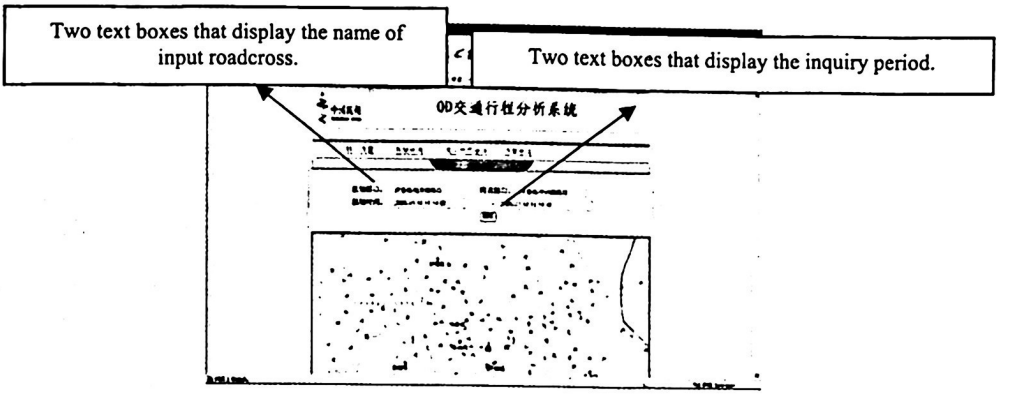


Fig. 2. The interface of graphic input

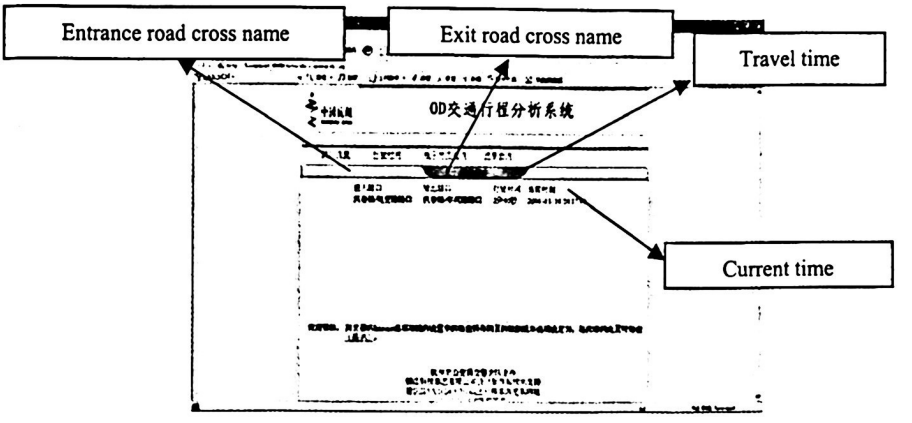


Fig. 3. Interface of Query Result

The approach has been put into practice in the intelligent traffic guidance information platform of Hangzhou, Zhejiang province, China. Results show that the method can provide travel time correctly and effectively.

Acknowledgment

This research was supported by the Important Project for Key Subject of Zhejiang Province, China 2006C13100 and the Projects of the Key Subject Open Fund of Zhejiang Province, China AMT200506-018 . The authors would like to thank Ms. WEN Xiao-yue and Mr. XU Jian-jun of the ZJUT-ENJOYOR ITS Joint Institute for their kind assistance in the research.

References

1. Zhenguo Zhou, Dynamic. OD Matrix Estimate and Utilization Research[D].Master's thesis of University Southeast,2001.
2. Chen Chaogang,Ma Zhu.The Improvement for The OD Investigation of Road Vehicle[J].EAST CHINA HIGHWAY,2003,140:69-73.
3. Luping Pan,Jingchang Chen.The Application of Static Mul-route Assignment Model on Freeway Plan[J]. Journal of Engineering Graphics.2003,3:37-42.
4. Taifeng Ma.Traffic Control and VehicleRoute Guidance in Intelligent Transport Systems[D].Doctor's thesis of Tianjin University,1999.
5. YangShao-hui, WangDai-hai,WangYing-ping,DongBin.Fitting Travel Time by Least Square Method[J].Joural of Highway and Transportion Research and Development,2006,9(23):90-93.
6. Wensheng Zhang,Qian Yin, Lixin Wu, Lei Xu,Zhigang Zang.The Prediction Model of Dynamic Rode Running Time Based on FCE[J]. Geography and Geo-Information Science,2006,22(4):25-27.
7. Cai Yonghua.The Research of OD Matrix in ITS[D]. Master's thesis of wuhan university of technology,2005.
8. Yong Lin,Yuanli Cai,Yongxuan Huang.Dynamic Origin-destination Matrix Estimation for Freeways[J].Journal of Chang'an University,2003,23(6):83-86.
9. Haihu Wang,Zhixiang Xu.Intelligent Traffic Monitoring and Measuring[J].TV Applications,2000.
10. M. Bierlaire, F. Crittin . An Efficient Algorithm for Real-Time Estimation and Prediction of Dynamic OD Tables[J] . Transportation Science EPFL, 2004,1(52).
11. Alan L. Erera ,Carlos F. Daganzo, David J. Lovell . The Access-Control Problem on Capacitated FIFO Networks With Unique O-D Paths is Hard . Transportation Science[J] ,2002,4(50):736-743.

Case-Based Route Planning in the Mexico City Subway System

Andrés Gómez de Silva Garza[1] and Lourdes Domenzain Ortega[1]

[1] Instituto Tecnológico Autónomo de México (ITAM), Río Hondo #1, Colonia Tizapán-San Ángel, 01000—México, D.F., México
{agomez, domen}@itam.mx

Abstract. The phrase "nearest-neighbor" is used in Case-Based Reasoning (CBR) to refer to a family of methods or heuristics for determining, during case retrieval, the similarity between a new problem to be solved and the previously-solved problems associated with the cases in a system's memory. In most problem-solving domains to which CBR has been applied, the concept of "nearness" is abstract, and determining whether a previously-solved problem is a "neighbor" to a new problem situation or not is done metaphorically. However, when the task to be performed involves route planning or related geographically-oriented activities, the original spatial connotations of "nearness" and "neighbor" come into play. In this paper we discuss the representation, storage, retrieval, and adaptation issues related to the measurement of nearness and the concept of neighborhood in case-based route planning systems in general. We also describe a particular system called SISBUSEC which plans routes in Mexico City's subway system and discuss differences and similarities with similar systems.

Keywords: case-based route planning, nearest-neighbor.

1 Introduction

The phrase "nearest-neighbor" is used in Case-Based Reasoning (CBR) to refer to a family of methods or heuristics for determining, during case retrieval, the similarity between a new problem to be solved and the previously-solved problems associated with the cases in a system's memory. In most problem-solving domains to which CBR has been applied, the concept of "nearness" is abstract, and determining whether a previously-solved problem is a "neighbor" to a new problem situation or not is done metaphorically. As an example, if the cases in a system represent the designs of buildings (*e.g.*, as in [1]) and a new set of problem requirements specified to the system include the desire to design a 30-storey office building in an earthquake zone, then the system might retrieve a case containing the design of a 35-storey office building in an earthquake zone. The values of both the *building-function* and the *ground-stability* parameters in the description of the new problem and in the retrieved case match identically, and perhaps all other earthquake zone office building cases in the case memory have a value smaller than 25 or greater than 35 for the *number-of-*

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 269–281

Received 25/05/07
Accepted 31/08/07
Final version 30/09/07

floors parameter, so the retrieved case is the "nearest neighbor" to the new problem situation available in the case base. However, none of these relevant, descriptive features (*building-function*, *ground-stability*, and *number-of-floors*) of the problem specification and design description have spatial values. Measuring the "distance" between two values of a numeric feature (even a non-spatial one) such as *number-of-floors* is straightforward (it simply involves finding the difference between the two values), but determining "distance" between different possible values of each of the non-numeric features might have to be done in an *ad hoc* manner.

For instance, with respect to the *building-function* parameter, domain knowledge might tell us that a value of *hotel* is closer to the desired value of *office-building* than a value of *parking-lot* is, for the purpose of determining the potential usefulness of a case. This domain knowledge would have to be incorporated into the part of the case retrieval process that measures similarity. "Similarity" is a more precise term than "distance" when discussing non-spatial features, though similarity is the inverse of distance: the nearest-neighbor case would be the one that is most similar, *i.e.*, least "distant," when compared to the new problem situation. All these observations apply to many other typical problem-solving domains that have been addressed from the standpoint of CBR, such as educational, medical and legal reasoning, aside from design tasks such as the one used above for illustrative purposes.

However, when the task to be performed involves route planning or related geographically-oriented activities, the original spatial connotations of "nearness" and "neighbor" come into play. Of course, the origin of the phrase "nearest-neighbor" is spatial in origin. One well-known algorithm for tackling the NP-complete Traveling Salesperson Problem (TSP) is known as the "nearest-neighbor heuristic" precisely because it involves having the "salesperson" cyclically decide to go to the nearest city (out of all the possible cities connected to the one he/she is currently in) at each step of the algorithm until all cities have been visited [2]. Prim's algorithm for finding the minimum spanning tree of a graph (with applications to minimizing the amount of gold/copper used in printed circuit design, for instance) also proceeds in stepwise fashion by finding the node of a graph that hasn't yet been visited nearest to the node that was last visited while constructing said tree [3]. These spatial uses of the phrase "nearest-neighbor" were eventually applied to more general, more abstract, non-spatial, but analogous situations such as the determination of similarity during the generic case retrieval process.

In this paper we discuss the representation, storage, retrieval, and adaptation issues related to the measurement of nearness and the concept of neighborhood in case-based route planning systems in general. We also describe SISBUSEC, a system that we have developed which performs case-based route planning in order to find routes within a city's subway system. The system currently has knowledge about Mexico City's subway, but we will be expanding it shortly. Mexico City's subway system at present includes 11 interconnected lines that contain a total of 174 stations (counting transfer stations multiple times, once for each subway line involved in the transfer, as they are physically different and involve a displacement in space from one line to the other, even if their names are the same) and total 202 kilometers in length. As we present the relevant research issues, we also mention related work that has addressed some of these issues, sometimes in similar ways to SISBUSEC and sometimes differently.

2 Case Representation

In general a new route planning problem might be specified by stating (or, in case the problem is being solved by an autonomous robot or vehicle, by sensing) the current location and stating a desired location (presumably different from the current one). We shall use the terms "origin location" and "destination location," abbreviated l_o and l_d, to refer to these points from now on. A solution to this type of problem would be a sequence of points l_0, l_1, l_2, l_3, ..., l_n that form a route of length n between l_o and l_d, where $l_0=l_o$ and $l_n=l_d$ for the route-planning problem that said sequence is a solution to. The intermediate points l_1, l_2, l_3, ... might represent landmarks or points of interest to look out for or pass through or next to, locations or intersections where a change of route, direction, or modality of transport must be performed, etc. In a CBR system, it is this sequence of points representing the problem solution, at the very least, that must be stored in a case for reuse in future route planning scenarios.

Assuming that a system is grounded in reality, the locations l_0, l_1, l_2, l_3, ..., l_n must be described as latitude and longitude or using some other method for specifying coordinates or positions within the universe of operation of the system unambiguously. Whether it is just the labels or pointers l_0, l_1, l_2, l_3, ..., l_n that are stored in a case, and the associated spatial domain knowledge stored elsewhere, or whether the actual values of the latitude and longitude that describe the locations are stored in the case, is a design decision that must be made in a case-based route planning system.

In SISBUSEC, locations represent subway stations and a case contains just the labels of the stations that form a route from l_o to l_d. This is because subway stations are likely to be repeated in several experienced routes (each stored in its own case), especially transfer stations. If the same value for l_o is given frequently in order to plan many routes over time from the same initial location (*e.g.*, one's home or office or hotel) to many others, then those stations near to l_o will also appear in several experienced routes. Because of this, in a separate database is where we store the actual latitude and longitude (and additional knowledge, described below) related to the different subway stations in a given city, and only the names of the stations are repeated from one case to the next if several cases' solution routes involve the same stations. In the Prodigy-based system described in [4], coordinates are also used to identify specific points (within the city of Pittsburgh). On the other hand, in the Router system [5], perhaps the most similar existing system to SISBUSEC, the symbolic names used for the streets of the Georgia Institute of Technology campus and the offices, corridors, and other functional areas of the College of Computing building don't have any additional associated information to indicate their position in "the real world."

3 Case Memory Organization and Case Retrieval

In general we can classify case retrieval algorithms broadly into two different types: nearest-neighbor matching and inductive methods [6]. For a spatial task domain such as route planning, given a new problem specifying l_o and l_d, nearest-neighbor

matching would compare l_o and l_d to the endpoints l_0 and l_n of the solution route stored in a case. A perfect match occurs if $l_0=l_o$ and $l_n=l_d$, in which case the solution route stored in the case can be directly presented as a solution to the new problem. In this situation, of course, the "nearest neighbor" is not really a neighbor, but rather the exact route that connects the two locations of interest. Fig. 1 shows, both textually and graphically, this situation in which a perfect match occurs between a problem specification and a case labeled *a*. For illustrative purposes, and without loss of generality, in this and the rest of the figures in this paper we will assume that retrieved routes are of length 4 and that the graphical representation of the solution is to be read from left to right (*i.e.*, the leftmost location shown is the origin location l_o, and the rightmost location is the destination location l_d).

Situation 1: $l_0 = l_{a0}$ and $l_d = l_{a4}$
Retrieved case: *a*
Retrieved case solution: l_{a0}, l_{a1}, l_{a2}, l_{a3}, l_{a4}
Desired solution = Retrieved case solution
$= l_{a0}$, l_{a1}, l_{a2}, l_{a3}, l_{a4}

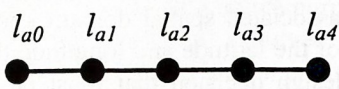


Fig. 1. Situation in which a perfect match occurs between a route-planning problem and the solution stored in a case.

A similar situation, also representing a perfect match but requiring the solution retrieved from the case to be reversed before presenting it as a solution to the new problem, occurs when $l_n=l_o$ and $l_0=l_d$. This occurs when the case contains a route from some location l_x to another location l_y and one asks a system to plan a route from l_y to l_x, and is illustrated in Fig. 2, which assumes that the match occurs between a new problem specification and a case labeled *b*. Note that this kind of situation only makes sense in route planning and perhaps a few other tasks in which reversing the components of a solution to a prior problem does not destroy the validity of the result as a solution to a new problem. On the other hand, it doesn't make sense to reverse previously-known solution routes to come up with new problem solutions in all route planning domains. In [4] reversal of routes stored in cases is not performed due to the prevalence of one-way streets in Pittsburgh. Unlike most subway lines around the world, one-way streets do not permit a traversal in both directions, so in this type of situation reversal is not a valid way to reuse the solution stored in a case.

The two situations described so far involve perfect matches, but what can a system do if there is no perfectly-matching case in memory? One solution is to give up, another is to use an alternate reasoning method instead of CBR, but a third is to retrieve the case that represents the closest-possible situation to the new problem, even if no perfectly-matching case is available. This is the type of situation in which nearest-neighbor retrieval (followed by case adaptation, which we consider below) comes into play.

Situation 2: $l_o = l_{b4}$ and $l_d = l_{b0}$
Retrieved case: b
Retrieved case solution: l_{b0}, l_{b1}, l_{b21}, l_{b31}, l_{b4}
Desired solution = Reverse (Retrieved case solution)
$= l_{b4}, l_{b3}, l_{b2}, l_{b1}, l_{b0}$

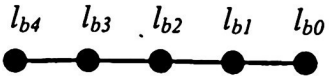


Fig. 2. Situation in which a perfect match occurs between a route-planning problem and the solution stored in a case, but the solution must be reversed before presenting it.

For route planning tasks, there are two different degrees to which one might be able to find a partially-matching case. First of all there might be a case in memory in which either l_o or l_d match one of the endpoints of the route stored in the case (but the other endpoint doesn't match—otherwise we would have a perfect match and thus one of the situations already described above). We will call this possibility P1. The second possibility, P2, is that neither l_o nor l_d match either of the endpoints of any of the solution routes in the cases stored in memory. In either situation, given a new problem specification there might be several partially-matching cases, and we would like the one that is nearest to the specification to be retrieved.

Let us assume that possibility P1 occurs, and that it's l_o that matches l_0 for a given case. If that's the only case for which l_o matches l_0, then that case is the nearest neighbor to the new problem situation and should be retrieved. However, if there are several cases for which l_o matches l_0, then some sort of tie-breaker criterion has to be implemented. The most logical tie-breaker criterion for a spatial domain such as route planning is to retrieve the case in which the other endpoint, l_n, is least distant from l_d in comparison to the other endpoints of the rest of the partially-matching cases. This case would be the nearest neighbor to the new problem situation. In the Router system, since no real positions are associated with street intersections (which is the type of location from or to which the system can plan routes), the tie-breaker criterion used is to select the first partially-matching case found. Sometimes this might turn out to be closest to the desired location than any other candidate partially-matching case, but most times it won't be.

Measuring the distance from l_n to l_d is not difficult assuming that these are not just labels but have actual positions (whether latitude and longitude, or some equivalent specification) associated with them, as in SISBUSEC (though the positions are not stored in the case, as mentioned earlier, but rather in a separate database of spatial knowledge). The nearest neighbor is the partially-matching case for which this distance is the smallest, out of all the potential candidate cases. If possibility P2 occurs, then the nearest neighbor is the case, out of all the potential candidates, for which the sum of the distances from l_o to l_0 and from l_n to l_d is the smallest.

If possibility P2 occurs it means that neither l_o nor l_d match either of the endpoints of any of the solution routes in the cases stored in memory, which means that finding

the nearest neighbor implies examining the entire case base once more (because all cases are potential candidates) after having already examined it to determine that there were no fully-matching or possibility-P1-type partially-matching cases, before deciding which case is nearest to the new problem situation. If the case base is large, this would be a time-consuming process. This is what might happen with SISBUSEC, since the Mexico City subway system has a total of 174 stations, and thus there could potentially be 174*173/2=15051 different cases (because this is the total amount of different problems that could be posed to the system) in the case memory.

This is the kind of situation that has led to the second general type of case retrieval in CBR, the inductive retrieval mechanisms. For case bases which are too large to make it feasible or logical to examine every single case to find how well it matches a new problem situation, inductive retrieval implies pre-classifying the cases into subsets. In some domains such as architectural design tasks [1] these subsets might overlap (a case may belong to several subsets), whereas in other domains it may not make sense to do so. Associated with the subsets of cases are decision-tree-like structures or procedures which provide indexing criteria by which a system can narrow down the search for a matching (or partially matching) case to a subset of the cases in memory. These structures provide an internal organization to a system's case memory, which may otherwise just be a "flat" list-like structure grouping cases together in no particular order (or perhaps in the order in which they were coded or learned, or some other arbitrary ordering that doesn't reflect the differing degrees of similarity among the cases).

In spatial reasoning domains the most natural way to group cases into subsets in order to achieve faster retrieval times is to use the concept of "neighborhood." If the geographic neighborhoods of l_o and l_d are known (or can be easily determined), and cases are somehow grouped together (classified) according to the neighborhoods that the endpoints of the routes they contain belong to, then a large number of cases can quickly be discarded without wasting time comparing them to the new problem situation during retrieval. If either a large number of neighborhoods or a large number of cases per neighborhood (both of which produce the same problem of having to make too many comparisons during retrieval, as before) result from a "flat" subdivision into neighborhoods, then another solution may be necessary. One possibility is for a hierarchical subdivision into neighborhoods and sub-neighborhoods into as many levels as needed. All of this, again, contrasts with non-spatial domains in which subsets of cases that are grouped together according to classification criteria that depend on the domain, whether the subdivision is hierarchical or not, define only abstract "neighborhoods."

In SISBUSEC, as mentioned earlier, we have a database which contains the latitude and longitude of each subway station, but it also contains knowledge about the neighborhood(s) that the stations belong to (and the connections between the stations). In order to improve the quality of the solutions produced by the system, the borders between neighborhoods in SISBUSEC are fuzzy, so stations lying on a border have more than one neighborhood associated with them. This also occurs with the Router system. In Router, however, the neighborhoods were constructed in an *ad hoc* manner, with varying numbers of neighborhoods at each level of the hierarchy and varying density of streets (or equivalent) in each neighborhood, whereas SISBUSEC

divides a geographical space into neighborhoods in a more systematic manner. The subdivision algorithm (described in the next paragraph) can therefore be easily automated in order to incorporate a new geographical area into the system without having to designate the neighborhoods of locations manually, as we did for Mexico City. The R-Finder system [7] has a two-level "hierarchy" of neighborhood-like areas referred to as "grids," into which they have divided the city of Singapore, and some streets may be contained in more than one grid in a somewhat similar fashion to SISBUSEC's and Router's use of fuzzy neighborhoods. The R-Finder "hierarchy" contains only two levels but is similar to Router's in the sense that it is a hierarchy of detail: at the highest level it is only the major roads and highways of Singapore that are represented, and at the lowest level (the grid level, where grid borders are defined by the major roads) is where the details about local roads, streets, and alleys are included. For a medium-sized city like Singapore, two levels might be enough, but if it were necessary to connect R-Finder to a similar system that contained knowledge about several Malaysian cities and the roads that interconnect them (and connect them to Singapore), or if it were necessary to group the roads in Singapore into more categories than just "major roads" and "all other roads" in order to make route planning more efficient, then more hierarchical levels may be necessary. The system described in [8] only subdivides its geographical universe of operation in the sense that it is split into cells which identify the possible initial and destination (and intermediate) locations.

SISBUSEC's method for defining neighborhoods is the following. A geographical space is divided hierarchically into quadrants (*i.e.*, into four subparts, no more and no less), producing what is known in geographical information systems [9], where it is a structure that is used frequently, and in computational geometry, as a "quad tree" [10]. Subdividing into smaller or larger amounts of neighborhoods might introduce north-south and/or east/west asymmetries into the resulting hierarchy and would not take advantage as fully of the idea of separating, at each node in the hierarchy, into a "reasonable" number of subparts. Each quadrant in SISBUSEC is recursively subdivided into quadrants if the situation merits it. This occurs when the locations in each resulting quadrant remain connected to each other within the quadrant and when the resulting quadrants end up having connections between them (otherwise the resulting "quadrants" are not really neighborhoods but very small sets of locations isolated from the rest of the locations in the geographical space). Fig. 3 shows, as a flat "map," one possible configuration of a geographical area after subdividing it in this quadrant-based fashion. Fig. 4 shows the hierarchy of neighborhoods and sub-neighborhoods that the subdivisions shown in Fig. 3 represent.

In SISBUSEC it is the leaf neighborhoods of each subway station that are stored in the database of spatial domain knowledge, since the hierarchy only serves to subdivide the geographical space. In contrast, in Router the hierarchy does not just subdivide the geographical area, but introduces more detail as one descends in the hierarchy (certain streets and therefore certain intersections are not mentioned at the root node, but appear in one or more of the neighborhoods at the next-lowest level due to their more local importance, etc.).

Returning to the issue of retrieving cases, due to the possibility that only partially-matching cases for a given new problem may be available, the potentially large size of the case base, and the spatial nature of the system's task, in SISBUSEC we have come

up with a case representation that allows us to combine aspects of nearest-neighbor and inductive retrieval. Our case memory is flat in the sense that it is just one database table (rather than several, each representing a subset of the cases) in which each row represents a given case, which would normally imply that nearest-neighbor retrieval is performed.

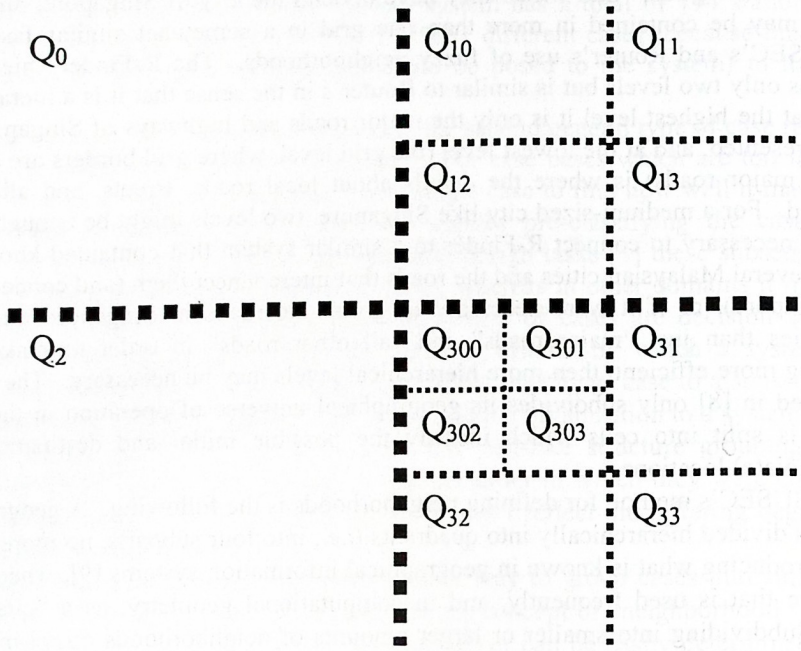


Fig. 3. Quadrant-based subdivision of a geographical space into neighborhoods and sub-neighborhoods.

On the other hand, the attributes with which we represent each case include not just the solution route, but also the origin and destination locations involved in the route, as well as their leaf-level neighborhoods, as distinct database attributes. This allows us to use l_o and l_d as primary indices during case retrieval, but also to use q_o and q_d (the quadrants or neighborhoods to which l_o and l_d belong) as secondary indices in case no perfect match occurs. The data that is retrieved if there is a match between a new problem situation and the primary and/or secondary indices of a case is the other attribute of the case: the solution (route) it contains. All of this is done without explicitly organizing the case memory into subsets of cases hierarchically. This contrasts with Router's approach, in which probing the case memory implies navigating down a hierarchical structure that reflects the organization of neighborhoods in its domain until a fully- or partially-matching case is retrieved or until a leaf neighborhood has been reached without finding a relevant case. Using q_o and q_d as secondary indices means that even if only partially-matching cases are found, they won't be just any random cases from memory, but rather they will be in the neighborhood (both literally and figuratively) of being good solutions. If no match is found between a new problem situation and any case in memory when using

even the secondary indices to probe the case memory, then it means that the system doesn't have any experience that is close enough to the new problem to be able to provide even a partially-useful solution. Case retrieval (or finding out that no cases were retrieved) in SISBUSEC is thus the result of performing between one and three quick database queries, depending on whether a perfect match, or partial matches of types P1 or P2 (as described above), occurred. Fig. 5 shows the case representation scheme used in SISBUSEC which supports this type of retrieval (each row in the database that contains the case memory describes a case in this way).

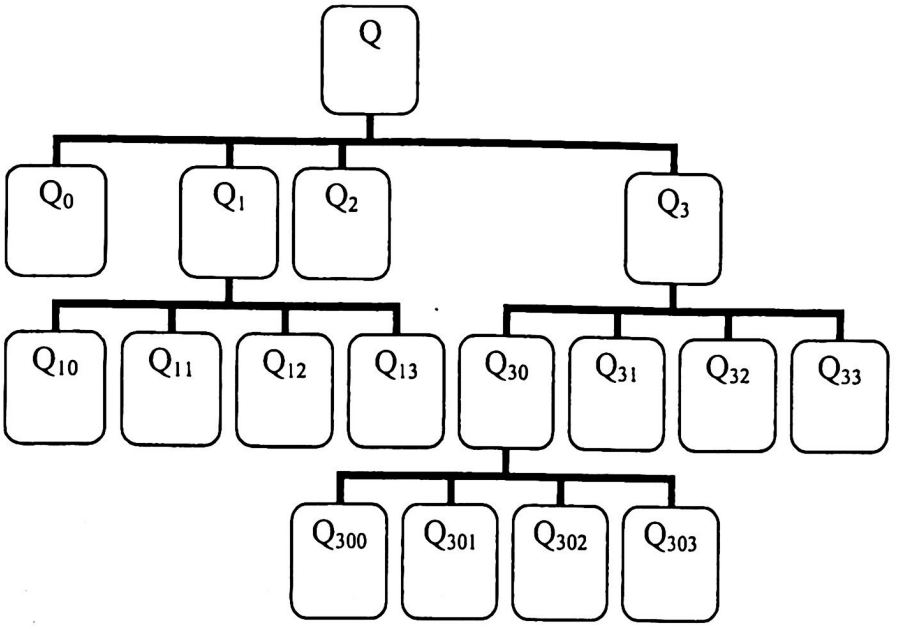


Fig. 4. Hierarchical (quad-tree) representation of the neighborhoods and sub-neighborhoods shown as a map in Fig. 3.

4 Case Adaptation

If no perfect or partial matches were found in the case memory for a given new problem, then CBR is not a feasible way to provide a solution to the problem. In these situations, SISBUSEC relies on the A* search algorithm [11], using straight-line distance as a heuristic function, to look for a solution using the graph which is implicit in the database that stores the domain knowledge about the locations and connections between the subway stations. This is similar to what happens in [4], [5], and [7], although Router's alternate algorithm is a heuristic map-based ("model-based") search that is not based on A*. If only partial matches were found during retrieval, then one of the retrieved cases must be adapted. There are several possible

situations depending on whether it was possibility P1 or P2 (described above) that occurred during retrieval. Whichever possibility occurred, what we know is that both l_o and l_d are either identical to or in the neighborhood of the endpoints of the route contained in the retrieved case. Since they are at worst close to the required locations, then case adaptation can be performed simply by extending the retrieved case (appending a new route) at one or both ends as needed. Router also operates in this fashion, as well as the system described in [8]. In SISBUSEC, the new route(s) to append are obtained using the same A* search algorithm that can be used by the system when no cases are retrieved, but are local searches (within the neighborhood(s) of the endpoints of the retrieved route) and therefore short and quick to perform.

Case *i*:

Solution: $l_{i0}, l_{i1}, l_{i2}, \ldots, l_{in}$
Primary index: l_{i0}
Primary index: l_{in}
Secondary index: q_{i0}
Secondary index: q_{in}

Fig. 5. Case representation scheme used in SISBUSEC to support neighborhood-based retrieval.

If possibility P1 occurred, then either the origin location matched one of the endpoints of the retrieved case but the destination location only matched the neighborhood of the other endpoint (possibility P1a), or else the destination location matched one of the endpoints and the origin location only matched the neighborhood of the other endpoint (possibility P1b). Fig. 6 illustrates possibility P1a assuming that the retrieved case is labeled *c*. The thick line shows the retrieved solution, and the thin line represents the appended route segment resulting from case adaptation.

Situation 3: $l_o = l_{c0}$ and $q_d = q_{c4}$ (but $l_d \neq l_{c4}$)
Retrieved case: *c*
Retrieved case solution: $l_{c0}, l_{c1}, l_{c2}, l_{c3}, l_{c4}$
Desired solution = Append (Retrieved case solution, Path(l_{c4}, l_d))
$= l_{c0}, l_{c1}, l_{c2}, l_{c3}, l_{c4}, l_d$

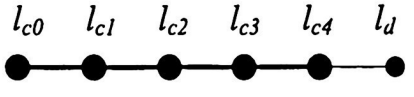


Fig. 6. Situation in which a partial match occurs between a route-planning problem and the solution stored in a case such that the origin location matches one of the endpoints of the route in the case, but the destination location is only close to the other endpoint.

A similar possibility (P1a') would occur if l_o matches the other (rightmost) endpoint, lc_4, of the retrieved case, instead of the leftmost one, lc_0, but we do not show this graphically. If possibility P1a' occurs, then case adaptation would involve reversing

the retrieved route before appending a new route segment to its right. We do not show possibility P1b graphically. A similar possibility, which we also do not show graphically (P1b'), would occur if l_d matches the other (leftmost) endpoint, l_{e0}, of the retrieved case, instead of the rightmost one, l_{e4}. This would mean that case adaptation would involve reversing the retrieved route before appending a new route segment to its left.

Assuming that neither the origin nor the destination locations were found to match any case in memory, but at least one case was found for which the neighborhoods of its endpoints match those of l_o and l_d, then possibility P2 occurred during case retrieval. This can again be broken down into two situations, P2a and P2b. In both situations, the retrieved route must be extended during case adaptation by appending new routes to both endpoints of the route obtained from the retrieved case (instead of only one endpoint, as in all the variants of possibility P1). In possibility P2a, the origin location's neighborhood matches the neighborhood of the leftmost endpoint of a case (and the destination location's neighborhood matches the neighborhood of the rightmost endpoint of the same case). In possibility P2b, the origin location's neighborhood matches the neighborhood of the rightmost endpoint of a case (and the destination location's neighborhood matches the neighborhood of the leftmost endpoint of the same case). In P2b, the retrieved route must be reversed before appending the two new routes. We do not show either of these two possibilities graphically, as they are quite similar to the situation shown in Fig. 6, except that the involve adaptation at both ends of the retrieved solution path instead of just one.

This method of adapting the route in a partially-matching case by appending one or two extra route segments to its endpoints can potentially lead to routes that contain repeated route segments. This is illustrated in Fig. 7, which assumes that the retrieved case is labeled h. In this example, it is assumed that the origin location in the new problem specification matched the leftmost endpoint of the retrieved case (*i.e.*, $lh_0=l_o$), but the destination location was only close to the rightmost endpoint (*i.e.*, $q_{h4}=q_d$, but $l_{h4}\neq l_d$). It is further assumed that the best route from l_{h4} to l_d, as found by A* during case adaptation, passes through l_{h3}, which happens to be one of the intermediate points in the solution retrieved from the case, thus ending up with the l_{h3}-l_{h4} route segment appearing twice in the route resulting from adaptation.

The Router system, which uses the same principles as SISBUSEC to adapt cases, just leaves the solutions like they were produced by the case adaptation algorithms equivalent to those described above, so some solutions would be as illogical as that shown in Fig. 7. In SISBUSEC, additionally to the above algorithms the resulting route is examined for duplicate route segments which, if found, are eliminated before presenting the route as a final solution. Thus SISBUSEC would produce the route shown in Fig. 8 as the final solution to the route that results from the first phase of adaptation as shown in Fig. 7.

In contrast to the previous approaches to case adaptation, [4] adapts cases by concatenating parts of as many cases as needed (partial cases), so the elimination of duplicate segments is not necessary (as they wouldn't appear in the constructed route—their avoidance is programmed into the case composition algorithm). A similar approach is followed by [12], though the emphasis in that system is the construction of a solution to one new problem associated with a given user from information related to several previous problems that might be stored in several case

bases, each of which belongs to (was generated by) a different user, *i.e.*, collaborative case-based route planning.

Retrieved case solution: l_{h0}, l_{h1}, l_{h2}, l_{h3}, l_{h4}
Path (l_{h4}, l_d) = l_{h4}, l_{h3}, l_d
Append (Retrieved case solution, Path (l_{h4}, l_d)) = l_{h0}, l_{h1}, l_{h2}, l_{h3}, l_{h4}, l_{h3}, l_d

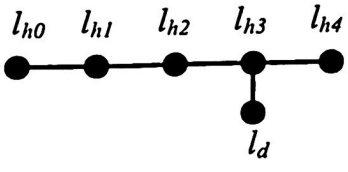


Fig. 7. Situation in which appending a route to the rightmost endpoint of a retrieved case (due to a partial match occurring during retrieval) results in a repeated route segment (and an illogical solution route).

Desired solution = RemoveDuplicateConnections (Append (Retrieved case solution, Path (l_{h4}, l_d)))
= RemoveDuplicateConnections(l_{h0}, l_{h1}, l_{h2}, l_{h3}, l_{h4}, l_{h3}, l_d)
= l_{h0}, l_{h1}, l_{h2}, l_{h3}, l_d

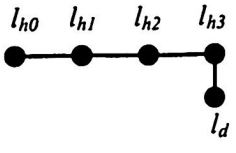


Fig. 8. Final solution route after eliminating the repeated route segments in the partial solution shown in Fig. 7 which results from case adaptation.

5 Case Learning

Since SISBUSEC can rely on the A* search algorithm to solve any problems in its problem space when a case is not available that can directly or partially help to solve the problem, then it can (and does) begin to operate even with an empty case base. This also occurs with [4], [5], [7], and [8]. When this happens, the system can store the solution as a case in order to rely more and more on these experiences, as more and more of them accumulate, when solving future problems in its domain of expertise. In SISBUSEC, this case storage/learning module takes into account the case representation and indexing scheme described above and illustrated schematically in Fig. 5.

6 Discussion

We have discussed many of the issues related to representation, storage, retrieval, and adaptation when using CBR for the task of route planning. We have also explained the decisions we have made, related to these issues, in order to implement SISBUSEC, a system that performs case-based route planning in order to plan routes within Mexico City's subway system. The name of the system is a contraction of a Spanish phrase meaning "sector-based search system" (sistema de búsqueda sectorial), from the fact that the concept of neighborhoods (quadrants, sectors) is central to the system's method for route planning. As we have shown, the concept of neighborhood has an impact in all aspects of the system's design: representation, storage, retrieval, and adaptation of cases. Finally, we have shown how some of the implementation decisions in SISBUSEC are similar and some are different from those made during the construction of similar systems that use CBR for route planning.

References

1. Gómez de Silva Garza, A., Maher, M.L.: Design by Interactive Exploration Using Memory-Based Techniques. Knowledge-Based Systems Vol. 9 No. 3 (May 1996), pp. 151-161.
2. MacGregor, J.M., Chronicle, E.P., Ormerod, T.C.: A Comparison of Heuristic and Human Performance on Open Versions of the Traveling Salesperson Problem. The Journal of Problem Solving Vol. 1 No. 1 (Fall 2006), pp. 33-43.
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms (Second Edition). The MIT Press, Cambridge, Massachusetts (2001).
4. Haigh, K.Z, Veloso, M.: Route Planning by Analogy. In: Veloso, M.M. and Aamodt, A (eds.): Case-Based Reasoning Research and Development: Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95). Lecture Notes in Computer Science, Vol. 1010 (1995), pp. 169-180.
5. Goel, A.K., Ali, K., Donnellan, M., Gómez de Silva Garza, A., Callantine, T.: Multistrategy Adaptive Path Planning. IEEE Expert Vol. 9 No. 6 (December 1994), pp. 57-65
6. Wang, X.: Basic Case-Based Reasoning Techniques. Chapter 2 of A Web-Based Case-Based Reasoning Tool, Master's Thesis, University of Wyoming (2000).
7. Liu, B.: Intelligent Route Finding: Combining Knowledge, Cases and an Efficient Search Algorithm. European Conference on Artificial Intelligence (1996), pp. 380-384.
8. Kruusmaa, M., Svensson, B.: Using Case-Based Reasoning for Mobile Robot Navigation. Proceedings of the Sixth German Workshop on Case-Based Reasoning (1998).
9. Burrough, P.A., McDonnell, R.A.: Principles of Geographic Information Systems. Oxford University Press (1998).
10. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications (2nd edition, revised). Springer-Verlag (2000).
11. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems, Science and Cybernetics Vol. 4 No. 2 (1968), pp. 100-107.
12. McGinty, L., Smyth, B.: Identifying the Routes Best Travelled by Collaborative CBR. Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science (2001), pp. 65-74.

Analysis of the Simulated Annealing Convergence in Function of the Standard Deviation and the Boltzmann Quotient for Scheduling Problems[1]

Martín G. Martínez-Rangel[1,2], Marco Antonio Cruz-Chávez[1], José Crispín Zavala-Díaz[2], David Juárez-Romero[1], Ocotlán Díaz-Parra[1]

[1]CIICAP-[2]FCAeI, Universidad Autónoma del Estado de Morelos
Avenida Universidad 1001. Col. Chamilpa, C.P. 62210. Cuernavaca, Morelos, México
{mmtzr,mcruz,crispin_zavala,djuarez,odiazp}@uaem.mx

Abstract. Simulated annealing (SA) converges by means of a probability of acceptance toward a minimum value of the cost function to a minimum temperature. When the cost function is very high, the probability of acceptance is minimum when temperature descends to a minimum value, for this, the probability is controlled for the temperature. An incorrect tuning of this parameter makes that the distribution of the probabilities of acceptance along the whole process of SA is slanted toward values very low or very high, what cause fall easily in local optimum. In this paper an analysis of correlation between the standard deviation and the distribution of probabilities of Boltzmann is made. The experimental results demonstrate that the standard deviation obtained through a sample of the solutions space of the problem, allow for a good tune of the initial temperature in SA.

1 Introduction

Many of the combinatorial optimization problems are classified like NP-hard, being JSSP (Job Shop Scheduling Problem) one of the most difficult problems to solve in this classification [2]. The required time to solve JSSP is increased exponentially according to the size of the problem. Benchmarks with 20 jobs and 20 machines are considered large because at the moment the solution is not know For great instances of these problems, there are no deterministic algorithms that can solve them. For this reason, they use a kind of metaheuristic nondeterministic that limits in polynomial time the approach to the global optimum in this set of problems [4].

Diverse metaheuristics have been proposed to the search of the global optimum. These include SA (Simulated Annealing) [5], [6], Tabu Search[7], [8], [9], Ant Colony [10] and Memetics Algorithms which are Genetic Algorithms used in a local search [10], [11], [12], [13] .The main characteristic that shares this algorithm is the great amount of time that us required to find or to approach a global optimum for very great instances by means of searches in neighborhoods of nondeterministic form. For JSSP, the metaheuristics work generates new solutions (Schedules) and review its

[1] This work was supported by Project 160 of the Fideicomiso SEP-UNAM, 2006-2007.

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 282–293

Received 24/06/07
Accepted 31/08/07
Final version 30/09/07

quality by means of evaluation of its scheduling, which assigns the times of beginning and ending for each one of the operations involved in the instance of the problem in order to be able to evaluate the objective function that is desired, for example the Makespan. The Makespan is defined as the time taken to complete the last O_i operation assigned in the process and is equal to the sum of the time of beginning of O_i plus the time of processing of O_i.

In this document, a new approach tune the temperature of beginning of SA help to converge or to approach faster to the global optimum for JSSP. The more critical problem presented SA it has to do with the tuning of its parameters, specially the temperature of beginning that is considered within SA like a control parameter, of the algorithm depends. There are other parameters not less important as the area quality of the generated solutions, the probability distribution for metropolis for the acceptance of new solutions, as well as the speed of cooling there are diverse approaches to tune SA. In [15] an approach uses a method that consists of the selection of an upper level in the temperature for each restart of SA. This upper level is selected when random way solutions are generated to take the best solution found in a time defined by test and error, the value of the upper level obtained when evaluating the objective function of the chosen solution. This improves the SA efficiency. In [17] an approach that the temperature fluctuates between a maximum and a minimum (feedback), the change of temperature will depend on a variable that can take positive or negative values (of random way) during each execution from SA.

For more than half a century, the mathematicians Bienaymé y Chebyshev [19] examined separately the property of the variability of the data around the average. They found at least that, regardless how a data set is distributed the percentage of observations that is contained within distances of plus-less k Standard deviations around the average must be $(1-1/k^2)100\%$ (rule of Bienaymé y Chebyshev). By means of an analysis of the behavior of the process of SA with respect to the distribution of the data and the application of this rule, in this document it is considered that the initial temperature is equal to twice the Standard deviation of solutions generated randomly, and that the analysis of the Boltzmann function to accept or to reject a solution will depend if the new solutions are in a rank (upper limit and lower limit) determined by the average of the generated solutions plus or less twice the Standard deviation. In order to accept as a matter of principle probabilistic 75% of all the solutions generated in the SA process, that is equivalent to 95% in processes that follow a normal distribution.

This work is composed of the following sections. Section 1 introduces the subject. Section 2 describes a generalized scheme of the algorithm of simulated annealing and the use of parameters. Section 3 proposes a new approach to tune the temperature parameters and the probability Boltzmann distribution function for the acceptance or rejection of new neighbors using the Standard deviation of the quality of the generated solutions there are at quality is also important and shows the algorithm of Simulated Annealing using this approach. Section 4 presents the computational study and the experimental tests using instances of problems known for JSSP. Section 5 presents the conclusions.

2 Generalized Scheme of Simulated Annealing

Simulated Annealing is a technique of stochastic local search approximates the minimum value of the cost function $f : S \rightarrow \Re$ on a finite set of S. It is an iterative method that moves in the space of solutions using a function of neighborhood N(x). When generating a new solution x´ of x, the solution candidate x´ is accepted as the new solution if $f(x') < f(x)$ or if $f(x') > f(x)$ is rejected or accepted in grasp of the function of probability of acceptance of Boltzmann P(x), which involves the parameter of control T, and the difference of the values of the quality of the solution (x´-x). Initially T has very high values and according to the algorithm progress. T decreases and influences in the probability of acceptance of the solution x´. The general procedure of SA [5] is defined in the following form:

1. Select a value of high beginning to T_0, a limit T_f to decrement to T_0 and an initial state x_0

   $T_t \leftarrow T_0$, $x \leftarrow x_0$
2. For each iteration k, k=1....k_f to do the following:
   a. Repeat until the balance is reached:
      i. Calculate the value of state x by means of the cost function: $E_k \leftarrow f(x)$
      ii. Generate a new state x′ using a neighborhood function, $x' \leftarrow N(x)$
      iii. Calculate the value of the state x′ by means of the cost function, $E_k' \leftarrow f(x')$
      iv. Assign $x \leftarrow x'$ according to the probability determined by the function of acceptance P(x)
3. Reduce T to $k+1$ using a control factor γ, $T_{k+1} \leftarrow \gamma * T_k$, where $0 < \gamma < 1$.
4. When T_{k+1} is less than T_f, finish
5. Return the best found solution x and their value of cost E.

On the whole the method of Simulated Annealing consists of a system of states x and the relations between the following functions

1. $f(x)$: A function of costs to be minimized

2. N(x): A mechanism of generation of neighborhood (it generates new states)

3. P(x): An acceptance function that decides if the new state is accepted or is rejected
4. T(k): A parameter of control of the annealing

For problems of numerical optimization, x is defined as a vector of integer or real numbers, and the function of Boltzmann P (x) is used for the acceptance of new states.

The Boltzmann distribution function uses a probability density function of the Gaussian type. This is defined as:

$$P(x') = \begin{cases} 1 & \text{if} \quad f(x') <= f(x) \\ e^{(-f(x')-f(x))/T} & \text{otherwise} \end{cases} \tag{1}$$

For JSSP, a state x is defined by a solution S (schedule) of the problem. The cost function f (x) is defined in this work by makespan C_{max} (S). Neighborhood N (S) of S is defined as a set of feasible solutions that can be generated from S by means of only one step. This step is a disturbance of a pair of operations (i, j) assigned in a M_j machine.

3 Parameters for Simulated Annealing using a dispersion measurement

The algorithm of Simulated Annealing requires that the used parameters have certain values which are determined generally a priori. Within the required parameters are included the parameters of control T_0 y T_f, the factor of temperature γ that defines the speed of decrement of T, as well as the length of the Markov chain which defines the number of iterations to carry out in the algorithm of Metropolis [25] in which some trades before generating a decrement in T. Each problem of scheduling has different characteristics, and consequently different degrees of difficulty. For this reason it is necessary to find the suitable values of the parameters involved in the process of simulated annealing. In order to tune SA, it is required that the indices witch allow knowing throughout the process if the values of the used parameters are the correct ones. Two of the indices are the uniform distribution $P(x)$ of the probability of acceptance of the Boltzmann distribution function (equation 3) and the distribution $P(n)$ at random n number generated and evenly distributed between (0,1) to determine the acceptance or rejection of a solution regarding the Boltzmann distribution function. Equation 2 indicates that the sum of the distribution of the probability of acceptance of the Boltzmann function must approximately be equal to the sum of the probability distribution $P(n)$ generated randomly (See figure 1). The problem obtained to achieve the equality as far as the uniform distribution of probabilities is that $P(e^{(-f(x')-f(x))/T})$ has a Poisson distribution, whereas $P(n)$ has a normal distribution. By probabilistic principle it is accepted that $P(e^{(-f(x')-f(x))/T})$ has a greater dispersion, this by the own behavior of the phenomenon of SA ,that at first accepts great variations of energy (very high probabilities) when T is very high, and in agreement T decreases and tends to zero, the variation of accepting by $P(x)$ is mini-

mum. Therefore the equation (2) is taken like a valued parameter which is composed of two indices, to evaluate the SA behavior (to see figure 2).

$$\sum P(e^{(-f(x')-f(x))/T}) \cong \sum P(n) \tag{2}$$

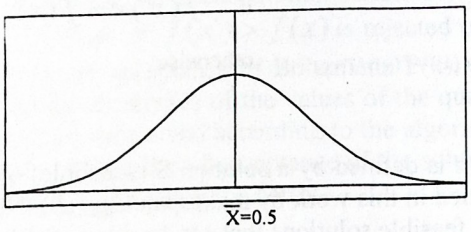


Fig. 1. Distribution of $P(n)$

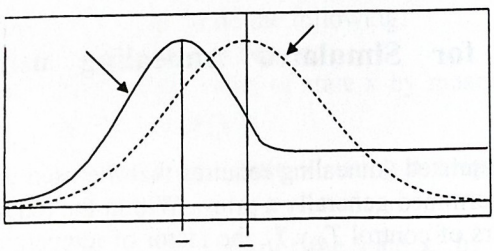


Fig. 2. Probability distribution of $P(e^{(-f(x')-f(x))/T})$ and $P(n)$

Aside from the equation 2 that shows the behavior of the SA process, it is another index that allows us to tune the parameters required in SA. This index is the standard deviation of the quality of the generated solutions that enter the process of SA within the procedure of Metropolis. This index shows us the dispersion degree that presents the variations in SA. In the statistic area, it is useful to know the degree dispersion that has the solutions in order to consider the degree of acceptance of these solutions. Regardless now a data set is distributed, the percentage of observations that are contained within distances $\pm$ k standard deviations around the average must be, at least:

$$(1-\frac{1}{k^2})100\% \tag{3}$$

Therefore, for the data whose polygons adopt any form, in the case of SA, at least [1 (1/22)]100%=75,0% of the observations must be contained within distances of 2 standard deviations $\pm$ around the average. 88,89% must be contained within distances plus or minus 3 standard deviations, and 93,75 standard deviations plus or less.

This is applicable when it is known that a particular random phenomenon did not follow the normal distribution pattern. In the case of SA, it follows a distribution pattern of Poisson.

In order to tune the degree of acceptance of a new solution, it is possible to chose between these three possibilities: for the SA process it is proposed to increase twice the standard deviation to the average value to obtain its upper bound and to diminish in the same proportion to obtain the lower bound (fig.3), which is equivalent to accept the 75,00% of the solutions in SA or the 95,44% in other processes with normal distributions. Table 1 shows the variation of the data to the around of the average according to the distribution pattern.

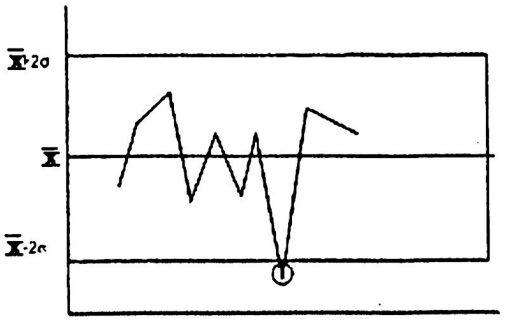


fig. 3. Minimum zone of occurrence to accept a 75% of the solutions for its evaluation in the proposed SA

Table 1. Variation of the data to the around of the average

Number of k units of standard deviation	Percentage of observations contained between average and k	
	Poisson distribution (Bienayme-Chebyshev)	Normal Distribution
1	No calculable	Exactly 68.26%
2	At least 75.00%	Exactly 95.44%
3	At least 88.89%	Exactly 99.73%
4	At least 93.75%	Exactly 99.99%

There are only accepted energy differences $\Delta E_k = -(f(x') - f(x))$, which are in the rank delimited by the $2*\sigma$, we can do that T_0 will be equal to the same $2*\sigma$ within the SA process ($T_0 = 2*\sigma$) in order that the distribution of $P(e^{(-f(x')-f(x))/T})$ will be uniform within SA. The importance to choose a good solution of beginning in SA to obtain good solutions already is mentioned in [15]. The values of each one of the parameters involved in the SA process based on the standard deviation are defined as follows:

1. For each one of benchmarks used in the tests, to generate a set of solutions Ω at random sufficiently to identify the standard deviation of the quality of the solutions that compose it, its average value, the minimum and maximum value found.
2. The Speed of cooling (length of the Markov chain), will be given by the double of large of the neighborhood of the problem [6] Ve = (n* (n-1))*2; where n is the number of jobs, which determines the cycle of Metropolis.
3. The value of *To* is equal to twice the Standard deviation ($2 * \sigma$) found for a set of solutions generated in point 1.
4. The decrement of the temperature is given by: $T \leftarrow \gamma * T$, γ is the coefficient of temperature that controls the cooling of the system.
5. A S′ solution of the problem could be evaluated to be accepted or to be rejected by means of the function of probability distribution of Boltzmann, as long as ΔE_k it will not be greater to twice the found standard deviation for the neighborhood upper bound and lower bound.

The previous parameters are used like an initial tuning to execute the process of simulated annealing. At a figure 4 shows the Algorithm of Simulated Annealing that allows to converge of faster way to optimal solutions for benchmarks of JSSP used like test, where $\Delta E_k = -(f(x') - f(x)) = \Delta S_k = -(f(S') - f(S))$ and $P(e^{(-f(x')-f(x))/T}) = P(e^{(-f(S')-f(S))/T})$.

```
1.  Input Data: S, σ
2.  Result: Makespan
3.  T = 2 *σ, Makespan = 10000;
4.  SizeNeighborhood ← Ve; //sampling size (n jobs)
5.  while (T > 0){
6.  Neighbor ← 0;
7.     while (NNeighbor< SizeNeighborhood){
8.        Generate a state S′ ∈ N ( S ) by means of a perturbation in S
9.        If (S' ≤ S)
10.       {
11.          S = S' ;
12.          if (Makespan > S) Makespan ← S;
13.          Neighbor++;
14.       } //end if
15.    else
16.       {
17.         ΔS ← S'-S
18.         if (ΔS ≤ 2 *σ) {
19.    With a randomly generated number α evenly distributed between (0,1)
20.            if ( α <exp (-ΔS /T))) Then {
21.               S ← S′ //accept new solution
```

```
22.                Neighbor++;
23.                } //end if
24.              else
25.                S ← S; //reject new solution
26.            }//end if
27.          else
28.              S ← S; //reject new solution
29.          } //end else
30.      } //end while
31.    T ← γ * T
32.  }// end while T_f > 0
```

Fig. 4. Simulated Annealing Algorithm with standard deviation

4 Computational Results

The proposed mechanism was implemented in C language in a PC with 2 GHz, and 1 Gb in RAM. In order to prove the efficiency of the proposed mechanism a set of benchmarks for JSSP of small size (Mt06, Mt10) [21], medium (La40) [22] and great (Yn1) [23] were used. The space of feasible solutions to be explored was generated by means of the neighborhood structure proposed in [20] and using the proposed mechanism of partial rescheduling in [24]. In table 2 there are the values of the standard deviation that were generated for the test instances. The set Ω is made up of 65.000 solutions for each one of the problems. In the reported results, the value of γ was fixed at 0.998, because it is the value at which the best results were obtained.

Table 2. Obtained standard deviation for Ω = 65000

Problem	Size	σ
Mt06	6x6	13.54
Mt10	10x10	153.63
La40	15x15	179.43
Yn1	20x20	108.91

The degree of dispersion of the solutions can be appreciated in figure 5. If to the obtained average they add plus or less twice for the σ, then we accept 75% of the solutions generated in the process of SA, that as it can be seen, the distribution pattern which it follows ΔE_k is similar to a Poisson distribution, where the greater number of optimal solutions is when ΔE_k it is approximated to zero (difference of energy between S' y S is minimum), reason why only good solutions when T_0 is minimum (it is approximate to zero) are accepted.

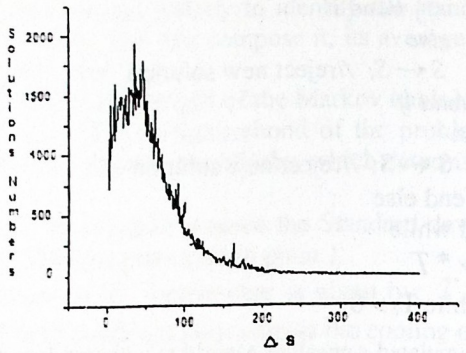


Fig. 5. Dispersion of the generated neighborhood for Yn1 20x20 problem

When limiting 75% of all the solutions within the space search of feasible solutions and tuning the initial temperature $T_0=2*\sigma$, it allows us to accelerate the convergence to a global optimum, when process of SA is controlled.

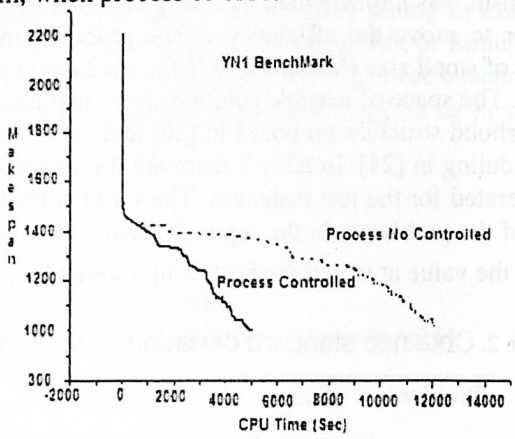


Fig. 6. Performance comparison using the Benchmark Yn1 20x20 problem

Figure 6 shows the results obtained for the Yn1 problem executing the algorithm of Simulated Annealing proposed using σ (controlled process) and to the algorithm of Simulated Annealing typical (without σ) or process no controlled, which initiates with an equal temperature to $2 * \sigma$ and accepting dependent solutions only by the criterion of Metropolis. The proposed algorithm manages to converge to makespan of 930 in 5526 seconds using like initial parameter of temperature $2 * \sigma$, and in addition, verifies that the new the proposed solutions as are within the fixed limits, that is to say, that $\Delta E_k \leq 2 * \sigma$. All the solutions were evaluated under this rule and her acceptance depended on the probability distribution of Boltzmann. The solutions that were outside the fixed limits were rejected automatically. In figure 6 one is in the controlled process that the time of convergence to obtain makespan of 1010 was of 4786 seconds, whereas in the no controlled process, the makespan was of 13960 seconds.

Table 3. Results for 4 instances of test for JSSP, using Controlled and Non-Controlled SA

Controlled SA						Non-Controlled SA		
Problem	Size Jobs x Machines	Best Know	C_{min}	RE	t_{min} Sec	C_{min}	RE	t_{min} Sec
Mt06	6x6	55	55	0.000	1200	57	3.6	5700
Mt10	10x10	930	930	0.000	2600	1010	8.6	7200
La40	15x15	1222	1240	1.4	4786	1278	4.8	12351
Yn1	20x20	886	930	4.9	5526	1010	13.99	13960

Table 3 show the best results of 5 executions for the 4 problems used as test, the data that appear corresponds to the relative error (RE) of the best solution (C_{min}) found and the required time to find this solution t_{min} for the four cases for the controlled and for non controlled process.

In this table it is observed that there is a great difference in the effectiveness of algorithms. The SA controlled in two small problems obtains optimal the global one and for the problems medium and great the relative error is of 1.4 and 4.9 respectively. These results contrast with the ones obtained by the non-controlled SA because the obtained relative error of 3.6, 8.6, 4.8 and 13.99 is much greater.

In order to show the efficiency of the controlled SA with respect to the non controlled SA, in figure 7, are shown the required times to obtain the same makespan in both processes. Analyzing the behavior of both SA one demonstrates that the efficiency of the SA is much better that in the no controlled SA, since little problems to larges problems (36 to 400 operations, for 6x6, 10x10, 15x15, and 20x20 problems).

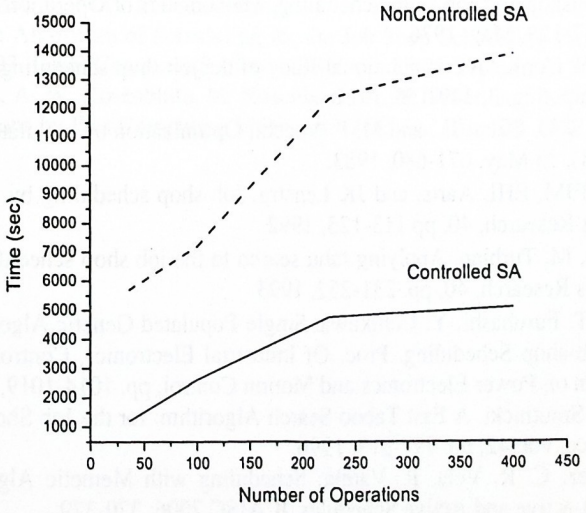


Fig. 7. Performance comparison using Mt06, Mt10, La40 and Yn1 problems

5 Conclusions

The tuning of the parameters used in SA represent the main challenge at the time of using this metaheuristics, The results observed in this document show that it is possible to tune the initial parameters for Simulated Annealing according to the degree of dispersion of the quality of the solutions, by means of the Standard deviation. Depending on the complexity of the problem, the values of the parameters change radically, reason why the Standard deviation turns out suitable like an average one to initialize the parameter of control for each kind of problem. The same Standard deviation is used to discriminate the solutions that will not be evaluated by means of the probability Boltzmann distribution function. The generated solution (S´) from a present solution (S) whose difference ΔS_k are greater to the $2^* \sigma$ is rejected and it is not evaluated as a feasible solution (this outside of range established by the inferior and the superior limit), this in order that $P(e^{(-f(S')-f(S))/T})$ is distributed of more uniform way. In the experimental test was observed that the mechanism, allows us to have a process of controlled SA, with a fast convergence to the global optimum.

References

1. A. Jain S. Meeran.: A State of the Art Review of JOBSHOP Scheduling Techniques. Technical Report. Department of Applied Physics, Electronic and Mechanical Engineering University of Dundee, Dundee, Scotland, K,DD1 4HN,1998.
2. M. R.Garey, D.S Jonson, and R. Shethi,; "The complexity of Flor Shop and Job Shop Scheduling, in Mathematics Of Operation research, vol. 1; No.2 (1976) pag. 117-129.
3. C H. Papadimitriou, K. Steigliths.: Combinatorial Optimization. Algorithms and Complexity. Dover Publications, Inc. 1998. [3] M.R. [3] Garey, D.S. Johnson and R. Sethi, The complexity of Flow shop and Job shop Scheduling. Mathematics of Operations Research, Vol. I, No 2, USA, 117-129, May, 1976.
4. D.Applegate, W. Cook, "A computational study of the job shop scheduling problem", ORSA Journal on Computing, 3, 149-156, 1991.
5. S. Kirkpatrick, S. D. Gelatt Jr., and M. P. Vecchi, Optimization by simulated annealing. Science, 220(4598), 13 May, 671-680, 1983.
6. V. Laarhoven PJM, EHL Aarts, and JK Lenstra. Job shop scheduling by simulated annealing. Operations Research, 40, pp.113-125, 1992.
7. M.D Amico M, M. Turbian. Applying tahu search to the job shop scheduling problem. Annual Operations Research, 40, pp. 231-252, 1993.
8. K. Morikawa, T. Furuhashi, Y. Uchikawa. Single Populated Genetic Algorithm and its Application to Job-shop Scheduling. Proc. Of Industrial Electronics, Control,Instrumentation, and Automation on Power Electronics and Motion Control, pp. 1014-1019, 1992.
9. E. Nowicki, C. Smutnicki. A Fast Taboo Search Algorithm: for the Job Shop Problem. Managemenr Science, vol. 42, pp. 797-813, 1996.
10. M. A. González, C. R. Vela, R. Varela: Scheduling with Memetic Algorithms over the Spaces of Semi-active and Active Schedules. ICAISC 2006: 370-379.
11. M. A. González, M. Sierra, C. R. Vela, R. Varela: Genetic Algorithms Hybridized with Greedy Algorithms and Local Search over the Spaces of Active and Semi-active Schedules. CAEPIA 2005: 231-240.

12.P. Moscato,. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms (Technical Report C3P 826). Pasadena, CA: Caltech Concurrent Computation Program, California Institute of Technology.
13.E.K. Burke and A.J. Smith, A Memetic Algorithm for the Maintenance Scheduling Problem, Proceedings of the ICONIP'97 Conference, Dunedin, New Zealand, 24-28 November 1997, (published by Springer), pp. 469-474.
14.P. J. Zalzala, and Flemming. Zalzala, A.M.S. (Ali M.S.),ed., Genetic algorithms in engineering systems /Edited by A.M.S. Institution of Electrical Engineers, London, 1997.
15.M. A. Cruz-Chávez, J. Frausto-Solís, D. J. Romero, Experimental Analysis in Simulated Annealing to Scheduling Problems when Upper Bounds are used, Transactions on Information Science and Applications, ISSN: 1790-0832, Vol. 2, Issue 5, pp. 581 - 586, 2005.
16.J. E. Beasley. OR-Library: Distributing test problems by electronic mail. Journal of the Operational Research Society,Vol. 41. No. 11, 1069-1072, 1990. Last update 2003.
17.Roy and Sussman, Les problemes d'ordonnancement avec contraintes disjonctives, Note D.S. no 9 bis, SEMA, Paris, France, December 1964.
18.M. Kolonko and M.T. Tran; Convergente of Simulated Annealing with Feedback Temperatura Schedule. Problem in The Engineeering and Informational Sciences, 11, 279-304, 1997.
19.M. G. Kendall, A. Stuart, The advanced Theory of Statistics, Vol.1 London: Charles Griffin 1958.
20.M. A. Cruz-Chávez, J.Frausto-Solís, J.R. Cora-Mora, Experimental Analysis of a Neighborhood Generation Mechanism Applied to Scheduling Problems, Proceeding of CERMA2006, IEEE-Computer Society, ISBN 0-7695-2569-7, pp 226-229, 26-29 September, México, 2006.
21.H. Fisher, and G.L. Thompson, Probabilistic Learning Combinations of Local Job Shop Scheduling Rules, In J.F. Muth and G.L. Thompson (Eds.), Industrial Scheduling, Prentice Hall. Englewood Cliffs. New Jersey 225-251. 1993.
22.S. Lawrence, Resource Constrained Project Scheduling: An Experimental Insvestigation of Heuristic Scheduling Techniques (Suplement). Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, Pensylvania, 1984.
23.T. Yamada R and Nakano, A Genetic Algorithm Aplicable to Large-Scale Job-Shop Instante Solving from nature 2, North-Holland, Ámsterdam 281-290, 1992.
24. M. A. Cruz-Chávez, M.G. Martinez-Rangel, J. A. Hernandez-Perez, J. C.-Zavala-Diaz, O. Díaz-Parra, An Algorithm of Scheduling for the Job Shop Scheduling Problem, Proceeding of CERMA2007, IEEE-Computer Society, ISBN, pp, 25-28 September, México, 2007.
25. N. Metropolis, A. W. Rosenbluth, M. Rosenbluth, A. H. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines." *J. Chem. Phys.* **21**, 1087-1092, 1953.

Natural Language Processing

Comparison of Three Algorithms for Automatic Keyword Extraction*

Mayra A. Paredes-Farrera and Hiram Calvo

Center for Computing Research, National Polytechnic Institute,
Av. Juan de Dios Bátiz s/n, esq. Av. Mendizábal, México, D.F., 07738. México
mparedesf0000@ipn.mx, hcalvo@cic.ipn.mx, www.hiramcalvo.com

Abstract: In this work, we present a comparative analysis of three algorithms for automatic keyword extraction: The first algorithm consists on simple frequency n-gram counts from the text; the second one uses Maximal Frequent Sequences to find relevant words, and the third one is an algorithm based on the complex network model: A complex network is built from bigrams in a text, then the density coefficients of this network for each node (word) are calculated. The relevant words are considered those with a higher density coefficient against a given threshold. The evaluation method used for comparing these algorithms is coverage, as defined in the Document Understanding Conferences (DUC). This coverage measure is determined by the number of overlapping words between the words selected as relevant by each algorithm and a summary model provided by humans (300 documents from DUC 2001). Our experiments show that the algorithm with better performance was Maximal Frequent Sequences.

Keywords: Automatic keyword extraction, Complex Networks, Maximal Frequent Sequences.

1 Introduction

Nowadays, the number of documents available electronically has grown considerably; searching, retrieving or indexing information in such a huge amount of text becomes then a complex task. The usage of keywords is an aid to make these tasks simple and fast.

Keywords within a document are relevant words that provide information about the content or the subject of such text. These words allow people to find in the pool of documents the information they are searching for. Search engines benefit also from the usage of keywords. When keywords are used, it is possible to obtain finer results in shorter time. A keyword can be a unique word (*bodies*), a morphologic root or stem (*bodi*), a lemma (*body*) or even a phrase (*The human body*, *the network*).

Extracting keywords manually might seem a trivial task if the number of documents is small, but for hundreds of documents, this task becomes tedious and costly.

* Work done under partial support of CONACyT, IPN (PIFI, SIP, COFAA), and SNI, Mexico.

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 297–304

Received 20/06/07
Accepted 31/08/07
Final version 21/10/07

Because of this, it is convenient to create algorithms to allow the extraction of these words automatically.

There are several techniques that implement the extraction of keywords in different ways. Usually implemented techniques are based on frequencies—count of the number of repetitions of words [9] . In some cases the count of frequencies is enough to fulfill some particular goal [8, 11, 17], but for certain task a more complex techniques are needed [10, 15].

Most of the recent techniques for automatic keyword extraction such as [7, 6 and 14] preprocess the text before extracting the keywords. This pre-processing consists of eliminating stop-words and phrases that begin or end with a stop-word; defining a set of PoS tag sequences; extracting all the words or sequences of words that match any of these; and even marking NP-chunks in the text. We are interested in performing automatic keyword extraction without the need of text pre-filtering, as this latter task would require resources which are language dependent.

In this work, we make a comparison between three techniques for automatic keyword extraction (extracting only one-word keywords). These techniques are based on n-grams (NGs), Maximal Frequent Sequences (MFS) [5] and Complex Network Models (CNM) [16].

Roughly, in NGs, all 2-grams 3-grams, up to 7-grams are extracted from the text. The obtained n-grams are ordered according to its frequency, and subsequently a threshold is used to decide which words are relevant.

For the algorithm based on MFS's, the algorithms first extract sequences from 2 to 7 n-grams from existing words in the text. To determine the relevant words, we define a minimum frequency threshold σ that each sequence in the document must surpass.

The CNM is based on concepts of complex networks following the notion that despite of its complexity, the human language can be represented with the use of graphs (networks). The words in human language interact in sentences and the co-ocurrence of words in sentences reflects language organization in a subtle manner that can be described in terms of a graph of word interactions [16]. CNM uses clustering coefficients of each word and extracts the relevant words that have a coefficient that exceeds a defined threshold.

The remaining of this paper is organized as follows: the next sections explain the algorithms to be evaluated: sections 2 to 4 describe the NGs, MSF and the CMN algorithms respectively, followed by the results of our experiments in Section 5. Finally, conclusions and future work are presented in Section 6.

2 N-Grams Algorithm

An n-gram is a contiguous sub-sequence of *n* items or elements from a given sequence. The underlying idea of the N-grams model (NGs) applied to text is to obtain all sequences of adjacent words (given a window size) which appear together within a text. Punctuation is not considered. A frequent sequence of words contains mostly relevant words for that particular text.

The window size of n-grams can be an arbitrary number ranging from 2 up to the size of the text. For n=2 we have bigrams, which are sequences of two adjacent

words; trigrams are sequences of three words and so on. In Figure 1 we show an example of the extraction of n-grams corresponding for an example text.

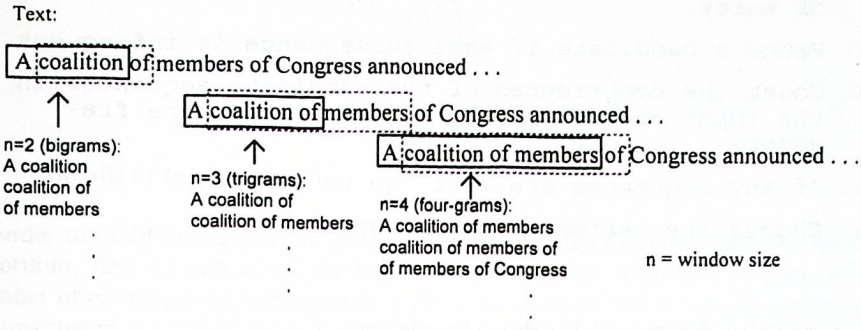


Figure 1. Example of adjacent n-grams extraction

3 Maximal Frequent Sequences Algorithm

For our purposes, we consider a Maximal Frequent Sequence (MFS) as a chain of words that appear often and are not contained in any other longer frequent sequence in a document collection. A sequence is considered to be frequent if it appears in at least σ documents, when σ is the frequency threshold given [2]. MFS are described by the following definitions:

Assume S is a set of documents, and each document consists of a sequence of words.

Definition 1 *A sequence* $p = a_1 \ldots a_k$ *is a subsequence of a sequence q if all the items* a_i, $1 \leq i \leq k$ *occur in q and they occur in the same order as in p. If a sequence p is a subsequence of a sequence q, we also say that p occurs in q.*

Definition 2 *A sequence p is frequent in S if p is a subsequence of at least σ documents of S, where σ is a frequency threshold given.*

Definition 3 *A sequence p is a maximal frequent (sub) sequence in S if there does not exist any other sequence p' in S such that p is a subsequence of p' and p' is frequent in S.*

The pseudocode of the Bottom-up algorithm [12] (taken from [1]) that we implemented to obtain the MFS is shown in Figure 2.

```
Input: A set of sequences, a frequency threshold
Output: A set of maximal frequent sequences

1. Collect all items of the input sequences, count them
   and select the frequent ones.
```

2. Build candidates sequences of length k+1 from frequent sequences of length k, where k is the number of words.
3. Prune a candidate if some subsequence is infrequent.
4. Count the occurrences of the candidate sequences in the input and select the sequences that are frequent.
5. If any sequences are left, go to Step 2.
6. Choose the maximal frequent sequences.

4 Complex Network Model Algorithm

The Complex Network Model (CNM) algorithm is based on concepts of complex networks and small-worlds. A complex network is a network (graph) with a non-trivial topological structure; the complex network built from a text presents small-world's (SW) characteristics The SW pattern can be detected from the analysis of two basic statistical properties: the clustering coefficient C and the path length d [16]. Antiqueira *et al.* [3 and 4] have modeled the text as a complex network. They show that it is possible to use this model to find solutions to diverse problems for natural language processing tasks.

In particular, TextRank [14] uses graphs for keywords and sentence extraction. The model we present here is similar to TextRank, but it is simpler in the sense that we do not consider the damping factor which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. In addition, we are not preprocessing the text (stop-word filtering, PoS tagging) because doing so would require language-dependent resources while we are searching for general language algorithms.

The graph for the language in a text (Ω_L), can be defined by $\Omega_L = (W_L, E_L)$, where $W_L=\{w_i\}$, $(i=1,\ldots, N_L)$ is the set of N_L words, and $E_L=\{\{ w_i, w_j \}\}$ is the set of edges or connections between words. $\xi_{ij} = \{w_i, w_j\}$, $(i,j=1,\ldots, N_L)$ indicates that there is an edge between words w_i and w_j, where $\xi_{ij}=1$ if a link exists and 0 otherwise. An edge is defined in this case as the existence of a bigram with frequency > 0 for the words w_i and w_j. The number of links per word is k. The set of nearest neighbors of a word $w_i \in W_L$ as defined by $\Gamma_i=\{j \mid \xi_{ij}=1\}$.

For the purpose of finding relevant words we calculated the clustering coefficient for each word. The clustering coefficient is defined as the number of connections between the words $w_i \in W_L$ as defined in equations (1) and (2).

$$L_i = \sum_{j=1}^{N_L} \xi_{ij} \left[\sum_{k \in \Gamma_i; j<k} \xi_{jk} \right] \quad (1)$$

$$C_v(i) = \frac{L_i}{\binom{\Gamma_i}{2}} \tag{2}$$

4.1 Details of Implementation

In order to build the network or graph for a text with the Complex Network Model algorithm, first we extract all the bi-grams from the text with their corresponding number of co-occurrences (frequencies). From the obtained data a non-directed graph is constructed. Each node from this graph represents a word, whereas every link of the node represents the adjacency of two words from each bigram. Each link is weighted by the number of times that the corresponding bi-gram appears in the text.

The clustering coefficient is calculated for each node (word) in the network and sorted in reverse order as show in Figure 4. Words with a clustering coefficient greater than 0 are considered relevant.

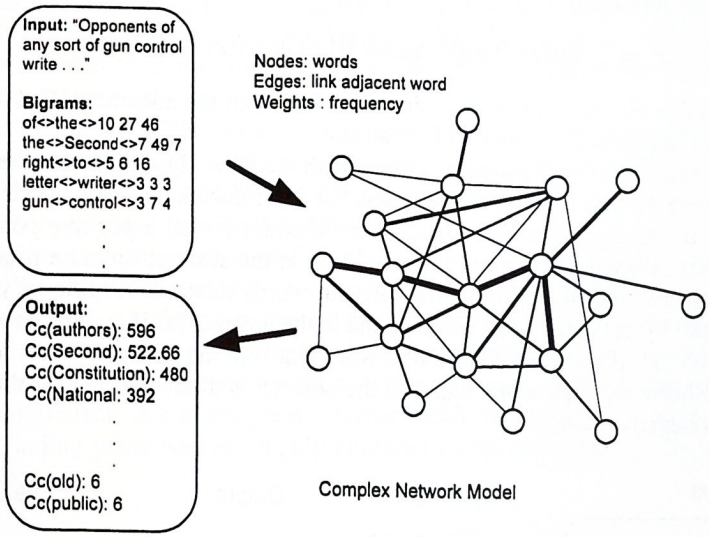


Fig. 2. Diagram of the CNM implementation. Darker links show a greater frequency of occurrence for particular bigrams in a text. *Cc*(*word*) is the clustering coefficient of *word*.

5 Experiments

We experimented with the available 300 documents from the Document Understanding Conference 2001 (DUC). The DUC is a series of evaluations of automatic text summarization systems organized by the National Institute of Standards and Technol-

ogy (NIST). Each document (body) has a summary model (abstract) written by humans; the documents topics are varied, as well as the size of each document. The average length of the documents is 762 words, ranging from 4580 words to 138 words.

Evaluation measure: The evaluation method we used for comparing these algorithms is *coverage*, as defined in DUC. This *coverage* measure is determined by the number of overlapping words between the relevant words chosen by each algorithm, and the abstract of each document. We rely on the idea that abstracts contain the most relevant keywords for a text. Precision is defined as a measure of the proportion of selected items that the system got right:

$$precision = \frac{tp}{tp + fp} \tag{3}$$

Recall is defined as the proportion of the target items that the system selected:

$$recall = \frac{tp}{tp + fn} \tag{4}$$

F-measure is a single measure of overall performance, combine precision and recall and as defined by:

$$F = 2 \cdot (precision \cdot recall)/(precision + recall) \tag{5}$$

We define our baseline as choosing all words from the document; that is, the intersection of the body document with its abstract.

The documents (body) were processed with the three algorithms and the results obtained were compared with the abstracts for each document. Each word selected as relevant by an algorithm and found in the abstract scored a positive point; whereas each word selected as relevant and not found in the abstract scored a negative point. This measured the **precision** of the relevant words selected. Additionally, we measured **recall** by comparing the words found in the abstract (as if it were a gold-standard for keywords). If the algorithm finds a word that is in the abstract, it scored a positive point. Otherwise, each word found in the abstract and not selected by the algorithm scored a negative point.

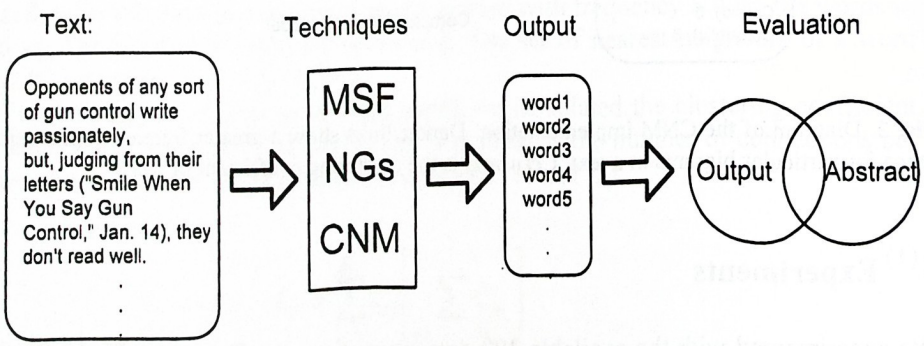


Figure 3. Experiment steps.

In Table 1, we show the results of our experiments obtained applying each one of the three algorithms to the DUC documents.

Table 1. Results of our experiments with the three algorithms.

Algorithm	Precision	Recall	F-measure
Body (baseline)	38.08	74.31	49.92
MFS	60.34	24.85	**34.16**
CNM	**62.92**	13.09	20.48
NGs	43.09	**25.01**	31.17

6 Conclusions and Future Work

We have presented a comparison between three techniques for automatic keyword extraction. Our results show that the algorithm with better performance considering both precision and recall was MFS. The baseline has the highest f-measure. It might seem that none of the algorithms was able to perform better than the baseline, however if we examine the precision column, we can see that the baseline has a very low precision. It is expected for most of the keywords from the abstract to be present in the body text. In this case, the coverage is high, but this also means that the keyword set contains many non-relevant words.

The algorithm using CNM has a lower combined performance; however it has the highest precision of all algorithms. This is because this algorithm obtains few but precise data. The results obtained by this algorithm are useful for certain applications which rely more on data precision than quantity.

Our experiment considered up to 7-grams. We shall experiment with greater n-gram sizes, as well greater window sizes to allow gaps in the n-grams.

As a future work we plan to evaluate directly with professional indexer keywords instead of comparing against abstracts. Another improvement is to consider the neighbors that are close to the each node, as well as the node itself for calculating the clustering coefficient; this might yield a higher recall. In addition we might experiment with finding paths between highly clustered nodes using algorithms based on random walks.

References

1. Ahonen-Myka, Helena, and Antoine Doucet, Data mining meets Collocations discovery. In Inquiries into Words, Constrints and Contexts, Festschrift for Kimmo Koskenniemi, pp. 194–203. CLSI Publications, University of Stanford, 2005.
2. Ahonen-Myka, Helena. Finding all maximal frequent sequences in text. In ICML99 Workshop, Machine Learning in Text Data Analysis. Bled, Slovenia, 1999.
3. Antiqueira L., M.G.V. Nunez, O.N Oliveira Jr, L.F Costa, Modelando Textos como Redes Complejas. In Anais do III Workshop em Tecnologia da informação e da Linguagem humana. TIL São Leopoldo-RS Brazil. July 22-26, 2005.

4. Antiqueira L., M.G.V. Nunez, O.N Oliveira Jr, L.F Costa. Strong correlations between text quality and complex networks features. Physica A, Volume 373, p. 811-820, 2007.
5. García-Hernández, René A., Jose Fco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa. A Fast Algorithm to Find All the Maximal Frequent Sequences in a Text. CIARP 2004, pp. 478-486, 2004.
6. Hult, A., Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction. Ph.D. Thesis, Departament of Computer and Systems Sciences. Stockholm University, April 2004.
7. Hult, A., Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on empirical Methods in Natural Language Processing, Japan, August, 2003.
8. Justeson, John S., and Slava M Katz. Co-ocurrences of antonymous adjectives and their contexts. Computational Linguistics 17:1-19, 1991.
9. Justeson, John S., and Slava M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Enginnering 1:9-27, 1995.
10. Kovács László and Helena Ahonen-Myka. Algorithm for Maximal Frequent Sequences in Document Clustering. 3rd International Symposium of Hungarian Researchers on Computational Intelligence. 2002.
11. Kupiec, Julian, Jan Pedersen, and Francine Chen. A trainable document summarizer. In SIGR '95, pp. 68-73, 1995.
12. Mannila Heikki, Toivonen Hannu, Inkeri Verkamo Discovering frequent episodes in sequences. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95), pages 210-215. Montreal, Canada. 1995.
13. Manning, Christhopher D., Shütze Hinrich. Foundations of Statistical Natural Language Processing. MIT Press, 2000.
14. Mihalcea Rada, Paul Tarau. TextRank: Bringing Order into Texts. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, July 2004.
15. Nguyen, Son N., Sun Xingzhi and Orlowska Maria E. Improvements of IncSpan: Incremental Mining of Sequential Patterns in Large Database. PAKDD 2005, pp. 442-451. 2005.
16. Ramon Ferrer I Cancho and Ricard V. Solé, The small world of human language, Proceedings of The Royal Society of London. Series B, Biological Sciences, 268(1482):2261–2265, 2001.
17. Ross, Ian C., and Jonh W. Tukey. Introduction to these volumes. In John Wilder Tukey (ed.), Index to Statistics and Probability, pp iv-x. Los Altos, CA:R & D Press, 1975.
18. Satanjeev Banerjee and Pedersen Ted. The Design, Implementation and Use of Ngram Statistics Package. In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, February 17-21, 2003.
19. Thiago Alexandre Salgueiro Pardo, Lucas Antiqueira Maria das Graças Volpe Nunes, Osvaldo N. Oliveira Jr. Luciano da Fontoura Costa "Using Complex Networks for Language Processing: The case of Summary Evaluation", 2006.

Homonymy Resolution and Tagging of Grammar Classes without Lexical Information using Neural Networks*

J. Ernesto Gómez-Balderas[1], Grigori Sidorov[1], Héctor Jiménez-Salazar[2]

[1]Center for Research in Computer Science,
National Polytechnic Institute,
Av. Juan Dios Batiz, s/n, Zacatenco, 07738,
Mexico City, Mexico
[2]Autonomous University of Mexico,
Mexico City, Mexico
sidorov@cic.ipn.mx, josegoba@sagitario.cic.ipn.mx,
hgimenez@gmail.com

Abstract. The paper describes the experiments that use artificial neural networks for resolution of the homonymy of parts of speech using limited contextual information, namely, only the information of parts of speech of the context is provided. We describe several experiments using backpropagation neural networks. Our experiments are conducted for Spanish language using LEXESP corpus. Two types of experiments are presented: the first one, when we are not aware of the real possible homonymy, and, thus, have to choose one of the fifteen possible grammar classes (in case of Spanish); and the second one, when we rely on the information obtained from the morphological analyzer, and resolve only the real homonymy presented in the text. For the first case we obtained more than 46% of precision, and for the second case more than 77%. Unlike other tagging methods we do not rely on lexical information. The advantage of this approach is drastic reduction of dimensionality of the decision space.

1 Introduction

The problem of part of speech tagging is one of the well developed problems in the area of computational linguistics. The problem consists in assigning of grammar tags to words in the text. There are different types of grammar tags, for example, they can include only parts of speech (grammar classes) information like nouns, verbs, etc. They can also include more detailed grammatical information. The part of speech tagging is very important in a lot of applications such as access to textual data base, robust syntactic analysis, etc.

Our purpose is to find a supervised method for tagging and resolution of morphological homonymy (it is equivalent to tagging with the subset of tags) that

* The work was done under partial support of National Polytechnic Institute, Mexico (CGPI, COFAA, PIFI) and Mexican government (CONACyT, SNI).

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 305–314

Received 15/06/07
Accepted 31/08/07
Final version 22/10/07

depends on distributional properties of tags in the text and does not depend on lexical information.

In this work, we apply the neural networks paradigm to POS tagging. Two types of the experiments are presented: the first one, when we are not aware of the real possible homonymy, and, thus, have to choose one of the fifteen possible grammar classes for Spanish; and the second one, when we rely on the information obtained from the morphological analyzer [8], and resolve only the real homonymy presented in the text, usually there are two or three tags for homonymic words.

The paper has the following structure. First, we briefly discuss the existing tagging methods, then we describe the preparation of the input data and experimental procedures for both cases (full tagging or homonymy resolution). After this, we present experimental results for various network structures and various weight assignments to different networks. Finally, conclusions are drawn.

2 POS Tagging Methods

There are different machine learning techniques that are used in POS tagging. For example, POS tagging with Hidden Markov Model (HMM) ([15]) involves counting occurences of the sequences and building a table of their probabilities. In advanced HMM learning, these probabilities are counted for triplets and larger sequences. Some other important current trends in POS tagging include Viterbi algorithm, Brill tagging algorithm, and Baum-Welch algorithm ([5], [6], [7], [10], [12], [13], [14]). All these approaches are dependent on lexical information, i.e., on coocurrences of individual words.

In [11], the distributional technique is applied to the similar formulated problems. The reported results for English have similar precision to ours, about 50%. In Spanish, the tagging problem using context is more difficult due to free word order.

On the other hand, neural networks ([1], [2], [4]) are also used in natural language processing. Martin Eineborg and Björn Gambäck ([3]) describe experiments with three different architectures of backpropagation networks for recognition of unknown words. Veronis and Ide ([16]) were using neural network approach for extracting lexical information from dictionaries. Another suggestion ([9]) is to use word structure for tagging with neural networks. Nakamura *et al.* ([17]) proposed prediction of words categories using neural network architecture called NETgram. This was also backpropagation approach. The grammatical category of preceding word was used to predict the next word category. We substantially extend these experiments.

3 Methodology of the Experiments

3.1 Input Data Preparation

The input data are obtained from the LEXESP corpus developed in Barcelona Polytechnic University. We had part of this corpus with manually resolved

homonymy, while the whole corpus is marked with all possible morphological tags, i.e, the homonymy is not resolved.

The first step was finding in the original corpus the phrases that were disambiguated manually (they were not subsequent phrases). Thus, we obtain the gold standard of the Spanish POS tagging. This data is used while calculating precision in automatic way.

The information in the corpus is encoded using the standard *Eagles* that is *de facto* encoding standard for Spanish language. For example, the word *alegre* (*happy*) has the following encoding: *AQ0CS0*, where *A* stands for adjective (also can be *N* for noun, *V* for verb, etc.), *Q* for qualifying, *C* for common (i.e., it can be both masculine and feminine), and *S* stands for singular.

At the next step, we prepared the corpus that contains only tags of grammar classes, i.e., the first letter of the code, as presented in Table 2 and Table 3. In our case, we always use only the first element of the tag that corresponds to the grammar class, say, *A* for adjective, and write it to a derived corpus of pure tags.

Finally, we prepared the input for neural networks that should contain only 1 and 0, because this is the only possible input of the networks. There are fifteen grammar classed in Spanish that we considered. Each tag from this file has an associated binary code containing only one 1 in the position of the tag in the ordering list, e. g., 001000000000000.

3.2 Neural Networks Training

We created eight networks depending on the number of context words. The difference between these networks is the number of the input neurons, see Table 1. Each network has fifteen output neurons that contain the final weights calculated for each grammar category. The number of input neurons depends on the number of contextual words (more specifically, their grammar categories) that are taken into account. Namely, it is the number of context words multiplied by number of possible grammar categories (fifteen in our case), for example, if we have two context words, then there are 30 input neurons, etc.

We used 1, 2 or 3 hidden layers in the neural networks in different experiments, see Table 4.

The following steps are made for training of these networks applying backpropagation approach.

Step 1:

Initialize weights connections with random values.

Step 2:

Present input patterns from the corpus. Depending on the context, the word was sent to the input of one or several networks. The networks were configured according to Table 1, where *P* is the desired tag in training and the unknown tag in recognition, and tags C_n correspond to tags of the context.

Step 3:

Calculate actual output for all networks.

Calculate input for neurons of the hidden layers on the basis of values calculated for the input neurons. For hidden neuron j, we have:

$$net_{p,j} = \sum_{i=1}^{N} w_{ji} x_{pi}$$

Calculate output for hidden neurons.

$$y_{p,j} = f_j(net_{p,j})$$

Calculate output for output neurons.

$$net_{p,k} = \sum_{j=1}^{L} w_{k,j} y_{p,j}$$

$$y_{p,k} = fk(net_{p,k})$$

Step 4:

Calculate the error for all neurons, starting from the output layer.

If k is a neuron from the output layer then the value is calculated as:

$$\delta^{o}_{p,k} = (d_{p,k} - y_{p,k}) f'^{o}_{k}(net^{o}_{p,k})$$

If neuron j is not from output layer then:

$$\delta_{p,j} = x_{p,i}(1 - x_{p,i}) \sum_{k} \delta^{o}_{p,k} w^{o}_{k,j}$$

Step 5:

Update weights starting from the output layer and going back to the input layer adjusting weights.

For weights of neurons in the output layer:

$$w^{o}_{k,j}(t+1) = w^{o}_{k,j}(t) + \Delta w^{o}_{k,j}(t+1)$$

$$\Delta w^{o}_{k,j}(t+1) = \alpha \delta^{o}_{p,k} y_{p,j}$$

For weights of neurons in the hidden layer

$$w_{j,i}(t+1) = w_{j,i}(t) + \Delta w_{j,i}(t+1)$$

$$\Delta w_{j,i}(t+1) = \alpha \delta_{p,j} x_{p,i}$$

Step 6:

This process is repeated until the mean error of the cycle is less than 0.001. $Ep = \frac{1}{2} \sum_i (d_i - o_i)^2$, where d_i is the element i of the desired output and o_i is the element i of the actual output.

3.3 Experimental Procedure for Tagging

The data from the LEXESP subcorpus, for which we have the gold standard, was divided into phrases. They were introduced into all possible networks, depending on the context of each phrase, e. g., if the phrase have thee words, and we are working with the word in the middle, then the networks 0, 1 and 3 participate in the processing, see Table 1.

For training, we introduced the tags of the context words into the corresponding neural networks and compared the results with the desired tag propagating the error backwards.

Table 1. Networks structure.

Phrase structure	Network
P-C_1	0
C_1-P	1
P-C_1-C_2	2
C_1-P-C_2	3
C_1-C_2-P	4
C_1-P-C_2-C_3	5
C_1-C_2-P-C_3	6
C_1-C_2-P-C_3-C_4	7

where P is a word we are working with, and C_n stands for the context words.

Table 2. Example of the input data.

Subphrases	Desired tag	Network
_V	Y	0
_VN	Y	2
_N	V	0
Y_	V	1
_NS	V	2
Y_N	V	3
Y_NS	V	5
_S	N	0
V_	N	1
_SN	N	2
V_S	N	3
YV_	N	4
V_SN	N	5
YV_S	N	6
YV_SN	N	7

where "_" stands for the tag that is currently processed (it is considered as unknown). It can be seen that the subphrase *YVN* was input for three networks 0, 1, 3 with input data *_V, Y_, Y_N*.

As usual in application of neural networks, the recognition is the same procedure only without propagation of error backwards. After training, we pass to experiments. In each phrase, we deleted one tag for experimenting, but its value was kept for comparison; this procedure was repeated for each tag. The network guesses correctly if the kept tag value was equal to the tag obtained in the network output. If more than one network was used, then we count the average value. We also tried different weights for different networks, see Table 5, Table 6, and Table 7.

A detailed example of the procedure is presented in Table 2. Suppose that have the tagged phrase *YVNSN*. It will be introduced into networks in the manner presented there.

The same phrases were used as input data for Brill tagger and TnT tagger for comparison with the proposed approach, see below.

3.4 Experimental Procedure for Morphologic Homonymy Resolution

We have at our disposal all possible tags for cases of morphological homonymy that occur in our corpus, as well as the gold standard of tagging, i.e., manual resolution of this homonymy. Our task is to choose one tag of the reduced set of possible tags. The difference with the previous case is that we do not have to use all tags, but only the tags that are homonymic in the real text. Namely, instead of fifteen tags, we have to choose between two or three possible tags.

The procedure is essentially the same as described in the previous section, being the only difference the restricted set of tags. For example, the homonymic tags are presented in Table 3.

Table 3.Homonymic tags.

Correct tags	Other possible tags
V	
T	N
N	TV
.	.
T	VM
N	

4 Experimental Results

In this section, we present experimental results obtained using previously described procedures.

4.1 Results for Tagging

We used 797,698 entries in the experiments. We experimented with the values of parameters like number of layers, number of iterations and learning factor for neural networks. The results presented in Table 4 were obtained, being the best performance 46.61%.

We compared our results with the results obtained using Brill tagger and TnT tagger. We understand that both taggers are not designed for the task as we formulate it, but, on the other hand, let us imagine that the language has only fifteen words (that corresponds to grammar categories), so the taggers that work with words can be applied as well. For Brill tagger, we obtained precision of 15.5%, and for TnT tagger only 6%. These are very low values that we explain with lack of information for correct performance of both taggers.

Table 4. Results using different network parameters.

Iterations number	Number of hidden layers	Learning factor	Precision %
250	3	0.25	46.38
150	3	0.25	46.30
150	2	0.5	**46.61**
100	3	0.25	42.80
100	2	0.25	46.37
100	2	0.5	46.01
100	2	0.75	46.10
100	2	0.9	45.78
100	1	0.9	46.28
100	1	0.5	46.16

4.2 Results for Morphologic Homonymy Resolution

The experiment for resolution of morphological homonymy was conducted for 1,830,349 entries that were used as input for different networks depending on the context size. We obtained precision of 77.08% using networks without any additional weighting, i.e., the results of all networks were considered equally important.

Still, another hypothesis is possible, that some contexts are more important than the others. For verifying this, we assign different weights to different networks at the moment of calculation of the resulting mean value. We assign weights in the following manners:

- Symmetric weights.
- Greater weights to the networks with major left contexts.
- Greater weights to the networks with major right contexts.

The results are presented in the following tables (Table 5, Table 6, Table 7). As can be observed, a preference to the right contexts gives a little bit better results, 77.45%.

Table 5. Results for symmetric weights.

Neural network weights								Precision
R0	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**%**
0.3	0.3	0.5	0.5	0.5	0.8	0.8	1	**77.36**
0.2	0.2	0.3	0.3	0.3	0.5	0.5	0.8	76.89
0.5	0.5	0.7	0.7	0.7	0.8	0.8	0.9	77.34
1	1	0.8	0.8	0.8	0.5	0.5	0.4	76.62
1	1	0.8	0.5	0.5	0.8	1	1	76.62

Table 6. Results for preference to the left contexts.

Neural network weights								Precision
R0	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**%**
0.2	1	0.2	0.5	1	0.5	1	1	75.71
0.5	1	0.5	0.8	1	0.8	1	0.8	**76.88**
0.5	1	0.3	0.5	1	0.2	1	0.5	76.45
0.5	1	0.3	0.5	1	0.2	0.8	0.5	76.38
0.5	1	0.5	0.5	1	0.5	0.5	0.5	76.53
0.2	1	0.2	0.5	1	0.2	0.5	0.5	75.70

Table 7. Results for preference to the right contexts.

Neural network weights								Precision
R0	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**%**
1	0.2	1	0.5	0.2	0.8	0.5	0.5	76.55
1	0.2	1	0.5	0.2	1	0.5	1	76.37
1	0.2	1	0.8	0.2	0.8	0.5	0.8	76.74
1	0.1	1	0.6	0.1	0.8	0.6	0.8	76.23
0.8	0.1	0.8	0.6	0.1	0.6	0.6	0.6	76.67
1	0.2	1	0.3	0.2	0.5	0.2	0.5	75.67
0.4	0.4	0.6	0.6	0.6	1	1	1	**77.45**

5 Conclusions

We presented experiments that use artificial neural networks for resolution of the homonymy of parts of speech (grammar classes) using the limited contextual information, namely, only the information of parts of speech of the context is provided.

We describe several experiments using backpropagation neural networks. Our experiments are conducted for Spanish language using LEXESP corpus.

Two types of experiments are presented: the first one, when we are not aware of the real possible homonymy, and, thus, have to choose one of the fifteen possible grammar classes (in case of Spanish); and the second one, when we rely on the information obtained from the morphological analyzer, and resolve only real homonymy presented in the text. For the first case we obtained more than 46% of precision, and for the second case more than 77%.

We tried various network architectures and various ways of weighting of different networks.

The advantage of the approach that does not rely on lexical information is drastic reduction of dimensionality of the decision space.

In future, it is interesting to calculate precision values for each grammar class separately.

References

[1] Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka. *Elements of Artificial Neural Networks*. The MIT Press, 1997.

[2] Russell D. Reed and Robert J. Marks. *Neural Smithing*. The MIT Press, 1999.

[3] Martin Eineborg and Björn Gambäck. "Tagging Experiments Using Neural Networks". In: R. Eklund (Ed.) *Proceedings of the 9th Scandinavian Conference on Computational Linguistics*, Stockholm University, Stockholm, Sweden, June 1993, pp. 71-81.

[4] B. Müller and J. Reinhardt. *Neural Networks*. Springer-Verlag, 1990.

[5] Eric Brill. Some Advances in Transformation-Based Part of Speech Tagging. In: *Proceedings of the twelfth national conference on Artificial intelligence* (vol. 1), USA, 1994, pp. 722-727.

[6] Atro Voutilainen. A syntax-based part of speech analyzer. In: *Proc. of 7th Conf. of European Chapter Assoc. Comp. Linguistics*, ACL, 1995, pp 157–164.

[7] Tetsuji Nakagawa and Taku Kudo and Yuji Matsumoto. Revision Learning and its Application to Part of Speech tagging. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL), Philadelphia, USA, July 2002, pp. 497-504.

[8] Francisco Velásquez, Alexander Gelbukh, Grigori Sidorov. AGME: Un Sistema de Análisis y Generación de la Morfología del Español. In: Julio Gonzalo, Anselmo Peñas, Antonio Ferrández (Eds.) *Proc. Multilingual Information Access and Natural Language Processing, International Workshop (November 12) at IBERAMIA-2002, VII Iberoamerican Conference on Artificial Intelligence*, Sevilla, Spain, November 12-15, 2002, p. 1–6.

[9] Helmut Schmid. Part-Of–Speech Tagging with neural networks. In: *Proceeding of COLING-94*, 1994, pp 172-176.

[10] Qing Ma, Masaki Murata, Kiyotaka Uchimoto, Hitoshi Isahara. Hybrid Neuro and Rule-Based Part of Speech Taggers. In: *Proceedings of the 18th International Conference on*

Computational Linguistics (COLING'2000), Saarbrucken, Germany, August, 2000, pp.509-515.

[11] Hinrich Schütze. Distributional Part-of-Speech Tagging. In: *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, Ireland, 1995, pp 141-145.

[12] Andrei Mikheev. Learning Part-of-Speech Guessing Rules from Lexicon: Extension to Non-Concatenative Operations. In: *Proc. of the 16th COLING*, Denmark, 1996, pp 770-775.

[13] Peter A. Heeman, James F. Allen. Incorporating POS Tagging into language modeling. In: *Proc. Eurospeech '97*, Rhodes, Greece, 1997, pp. 2767-2770.

[14] Simon Cozens, Pusey Lane. Primitive Part-Of-Speech Tagging using Word Length and Sentential Structure. *CoRR: Computation and Language*, 1998.

[15] Thorsten Brants. TnT A Statistical Part-Of-Speech Tagger. In: *Proceedings of the 6th Applied NLP Conference, ANLP-2000*, USA, April 29 -- May 3, 2000.

[16] Jean Veronis and Nancy Ide. Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries. In: *Proc. of Int. Conf. COLING*, Finland, 1990, pp. 1-6.

[17] M. Nakamura *et al.* Neural network approach to word category prediction for English texts. In: *Proc. of 13th Int. Conf ACL*, Finland, 1990, pp. 213-218.

Extract Conceptual Graphs from Plain Texts in Patent Claims

Shih-Yao Yang and Von-Wun Soo

Department of Computer Science, National Tsing Hua University,
HsinChu 300, Taiwan
yao@cs.nthu.edu.tw, soo@cs.nthu.edu.tw

Abstract. This paper develops techniques to extract conceptual graphs from a patent claim by using syntactic information (POS, and dependency tree) and semantic information (background ontology). Due to plenteous technical domain terms and lengthy sentences prevailing in patent claims, it is difficult to apply a NLP Parser directly to parse the plain texts in the patent claim. This paper combines such techniques as regular expressions, finite state machines, Part-Of-Speech tags, conceptual graphs, domain ontology and dependency tree, to convert a patent claim into a formally defined conceptual graph. From 100 patent claims, the average precision and recall of a concept class mapping from the patent claim to domain ontology are 96% and 87% respectively and the average precision and recall for ***Real*** relation class mapping is 97% and 98% respectively. For the concept linking of a relation, the average precision is 79%.

Keywords: Conceptual Graph, Natural Language Processing, Patent Document Analysis, Patent Claims Information Extraction, Ontology, Dependency Tree, Regular Expression.

1 Introduction

Information extraction (IE) from a plain text is not an easy task. Traditional information extraction method [1] extracts patterns, part of sentences, using regular expressions or pattern templates. [2] proposed a natural language processing method to extract the structure and chunk phrases of a patent claim. Both of them did not utilize semantic inferences from the point of view of either a complete sentence or a whole document. [3] proposed a method to analyze the rhetorical structure of a patent claim to improve the readability of the patent claim but did not consider the semantic relations among words in the patent claim. In this paper, we propose a method to extract conceptual graphs from the plain text of a patent claim based on the syntactic information (Part-Of-Speech tags and Dependency Tree) generated by a NLP parser and the semantic information derived from a background domain ontology. After information extraction, the patent claim sentences can be mapped into a single connected conceptual graph [4]. The conceptual graph can also be converted to a corresponding first-order-logic formula [5] in a way that the semantic inferences can be carried out further.

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 315–329

Received 14/06/07
Accepted 31/08/07
Final version 17/09/07

However, there are two main problems to build a conceptual graph based on the semantic information extracted from a patent claim: (1) A patent claim sentence is usually so lengthy as to cause a parsing crash; (2) It must have a systematic method to map a patent claim to its corresponding semantic structure, namely a conceptual graph in terms of a background domain ontology. In dealing with problem 1, the system analyzes the organization of a patent claim and uses a finite state machine to split a patent claim sentence into a set of sub-sentences before applying an NLP parser. In dealing with problem 2, we construct conceptual hierarchies for the domain ontology and map contents of a patent claim to the conceptual classes in the domain ontology in terms of Part-Of-Speech tags and the contents themselves, and connect the conceptual classes which have semantic relations in the patent claim by using the dependency relation information in the dependency tree generated by the NLP parser. Our ultimate goal of research is to develop automated approaches to conduct comparison and summarization over patent claims based on the extracted conceptual graphs.

2 Dependence Tree and Conceptual Graphs

2.1 Dependency Tree

A dependency tree [6] is a directed tree that is different from a phrase structure in representing a sentence. Each node in a dependency tree can be represented by a word from a sentence and each edge in a dependency tree may link two words by a syntactic relation. In contrast, each node in a phrase structure is either a Part-Of-Speech tag or a word from a sentence and each edge in a phrase structure may link two nodes without a syntactic relation between them. Both the phrase structure and the dependency tree for the sentence "particles comprise a smectite clay" where smectite is a type of the clay from USPTO [7] 7112123 generated by the Stanford NLP Parser are shown in Figure 1. A phrase structure takes into consideration among multi-word constituents, while a dependency tree considers the relationship between two individual words.

2.2 Conceptual Graphs

A conceptual graph is a kind of formal knowledge representation in terms of Semantic Networks and existential graphs. It is a bipartite graph which consists of two types of nodes, a concept node denoted as a rectangle and a relation node denoted as a circle, where only concepts can link with a relation. "In a conceptual graph, concept nodes can represent entities, attributes, states, and events, while relation nodes can show how the concepts are interconnected"[4]. A conceptual graph is related to a support which is the background domain ontology for a specific domain. This support can be formalized as:

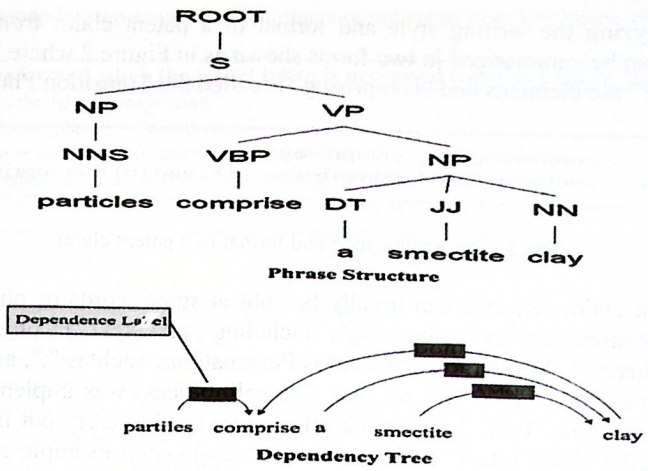


Fig. 1. A phrase structure vs. a dependency tree for the sentence "particles comprise a smectite clay".

1. A concept hierarchy which has the partial ordering operator over the concepts. For concept labels A, B, and C, if $A \leq B$ and $B \leq C$, then $A \leq C$.
2. A relation hierarchy which is disjoined from the concept hierarchy. Each relation links two or more concepts.
3. A set of star graphs. Each star graph consists of a relation and a set of concepts that the relation can link.

A set of individual markers that refer to specific entities. The generic marker "*" refers to an unspecified entity. If an individual marker conforms to type C, it must also conform to all supertypes of C. For example, John is an individual marker of the concept "Man", then it is also an individual marker of the concept "Person" which is a supertype of "Man".

3 Split a Patent Claim into a set of Sub-Sentences

To parse a patent claim using the NLP Parser, one must deal with the problems of a lengthy sentence in the patent claims. We split a long sentence into a set of sub-sentences and then parse each sub-sentence one by one and then combine the parsing results of each sub-sentence. However, to split a patent claim sentence into several sub-sentences, it must make sure that each sub-sentence is not only syntactically parsable but also concept preserving. By "parsable", it means the NLP Parser can parse the sentence in reasonable time. The maximal length of a sub-sentence is set at 70 words in the current application because it is a threshold above which the NLP Parser tends to crash. Concept preserving means the split would not lose or distort the meaning of the original patent claim. To ensure the two requirements, good heuristics of splitting are devised according to writing style and format of a patent claim.

By analyzing the writing style and format of a patent claim from a corpus of patents, it can be summarized in two forms shown as in Figure 2 where "A", "B", "C", "D", and "E" are elements and "comprising" is called the Transition Phrase.

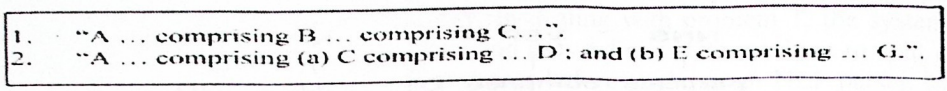


Fig. 2. The writing style and format of a patent claim.

A patent claim sentence can usually be split at some words or phrases: (1) The transition phrases, such as "comprising", "including", and etc; (2) Conjunction words, such as "whercin", "therefore", and etc; (3) Punctuations, such as ";", and "."; (4) List Items, such as "(a)", "(i)", "a)", and etc. The split process was implemented using a finite-state machine. Table 1 shows the actions that need to carry out in order to deal with a specific target token. Table 2 shows a step-by-step example of splitting the patent claim from US patent 6979252 shown in Figure 3. According the devised splitting heuristics, six parsable sub-sentences were obtained and none of them lose or alter their original meaning in the original sentence.

Table 1. Actions performed in dealing with a target token

Target Token	Actions performed on the target token

Transition Phrase (comprising, including, having, consisting of, ...)	1. Set transition_phrase = Target Token;
	2. left_string + transition_phrase + right_string
	Set sub-sentence = left_string + Transition Phrase + temp_word
	// temp_word is a dummy word
	3. Parse the sub-sentence and save the parsing result.
	4. Get the subject of temp_word from the parsing result.
	And add ("the" + subject) before the transition_phrase and continues the process.
	// The system utilizes the determiner "a" and "the" to determine
	// if a noun coming forth for the first time.

Connected-word (wherein, therefore, ...)	1. Set sub-sentence = the left string of Connected-word 2. Parse the sub-sentence and save its parsing result. 3. Continues the process.

List Item ("(a)", "(i)", "a)", ...)	1. Read the next token
	If (the next token is a VBG (Verb, gerund/present participle) or a VB (Verb, base from))
	Add "a step which" before the token
	End if

Punctuation (";")	The action is same as the action of Connected-word.

Table 2. A step by step example of splitting a patent claim from US patent 6979252.

Step	**Actions performed when the target token is processed** (Input is a patent claim)
1.	The target is the first "**comprising**" 1. Set transition_phrase = "comprising" *// left_string is the left of the current target token* 2. Set sub-sentence = left_string + "comprising" + "temp_word" "A method for separating and removing soluble polymeric silicates in a polishing slurry, said slurry comprising temp_word" *// in patent document "said" means "that", so we change "said" to "that" before parsing* 3. Parse the sub-sentence and save its parsing result. 4. The subject of "temp_word" is "slurry". Add "slurry" before the "comprises" and continues the process. *// reset the patent claim string* *//right_string is the right of the current target token* Set patent_claim = "the slurry" + "comprises" +right_string
2.	The target is the second "**comprising**" 1. Set transition_phrase = "comprising" 2. Set sub-sentence = left_string + "comprising" + "temp_word" "the slurry comprises a colloidal dispersion of silica, prior to a chemical mechanical polishing process, the method comprising temp_word" 3. Parse the sub-sentence and save the parsing result. The subject of temp_word is "method". Add "method" before the Transition Phrase and continues the process. *// change "comprising" to "comprises" to make it grammatical* Set patent_claim = "the method" + "comprises" + right_string
3.	The target is "**a)**" Read the next token "introducing". Because "introducing" is a VBG. add "a step which" before "introducing".
4.	The target is the first ";" 1. Set sub-sentence = the left string of ";" "the method comprises a step which introducing the polishing slurry into a centrifuge" 2. Parse the sub-sentence and save the parsing result.
5.	The target is "**b)**" Read next token "separating". Because "separating" is a VBG. add "a step_b which" before "separating"
6.	The target is the second ";" 1. Set sub-sentence = the left string of ";" "the method comprises a step which separating via centrifugation the soluble polymeric silicates, as a portion, from the polishing slurry to yield a product slurry" 2. Parse the sub-sentence and saving the parsing result.
7.	The target is "**c)**" Read next token "removing". Because "removing" is a VBG, add "a step which" before "removing" and skip "and" before "c)"
8.	The target is "**wherein**" 1. Set sub-sentence = the left string of "wherein" "the method comprises a step which removing the product slurry from the centrifuge" 2. Parse the sub-sentence and save the parsing result.
9.	The target is the end of patent claim 1. Set sub-sentence = current all token sequences "wherein the product slurry has a lower level of soluble polymeric silicates than does the polishing slurry" 2. Parse the sub-sentence and save the parsing result.

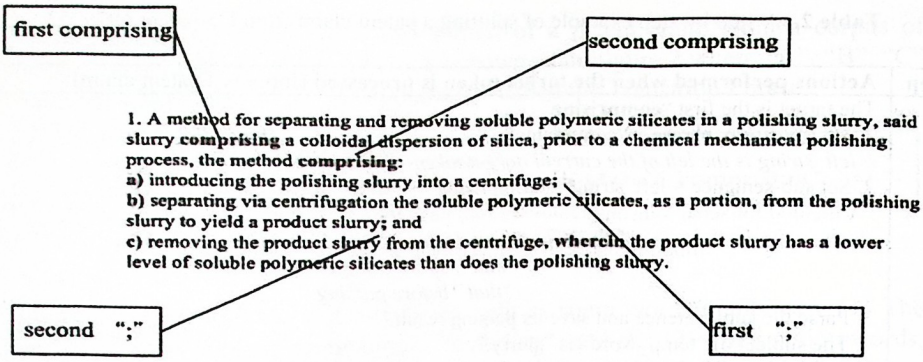


Fig. 3. A patent claim from US patent 6979252

4 Domain Ontology

The background domain ontology includes the concept and relation hierarchies, a set of star graphs that decide which concepts are neighbors of a given relation, and the markers of the concept and relation classes.

-Concept hierarchies

By analyzing the format and writing style of a patent claim, we constructed the high levels of concept hierarchy as shown in Figure 4.

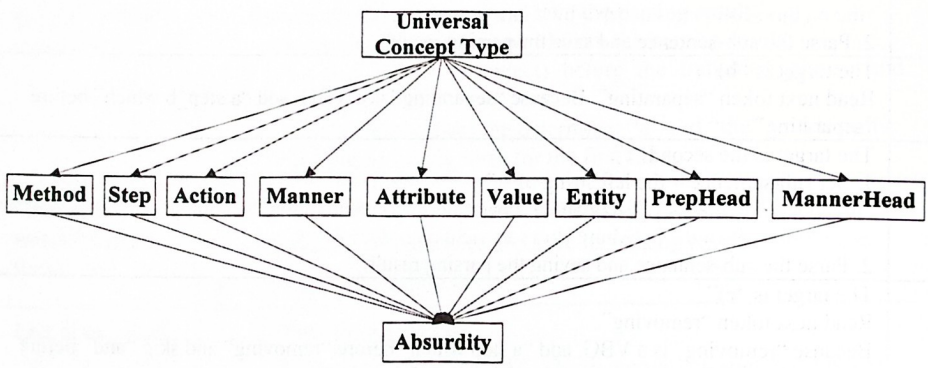


Fig. 4. The Concept Hierarchy of a Patent Claim.

- Relation hierarchies and Star graphs of the relation

We summarize totally 4 relations and Table 3 shows the degree of neighbors for each relation and the candidate concepts that the relation can link.

Table 3. A step by step example of splitting a patent claim from US patent 6979252.

Relation labels	Neighbors (Star graphs of the relation)	Description
Verb	(Entity, Entity, PrepHead, MannerHead);(Entity, SpecialPhrase, PrepHead, MannerHead);(Entity, Value, PrepHead, MannerHead);(Method, Step, PrepHead, MannerHead);(Method, Entity, PrepHead, MannerHead);(Method, Action, PrepHead, MannerHead);(Method, Entity, PrepHead, MannerHead)	The first neighbor represents a subject and the second neighbor represents an object of the verb. The third neighbor PrepHead is a link between relations Prep and Verb, and the fourth neighbor MannerHead is a link between relations AssAtt and Verb.
Contains	(Entity, Entity);(Method, Step);(Method, Entity);(Method, Action);(Method, Entity)	The first neighbor contains the second neighbor.
Prep	(Entity,Action);(Method,Action);(Step,Action);(Entity, Entity);(Method, Entity);(Entity, Value);(PrepHead, Entity);(PrepHead, Value);(PrepHead, Action);(PrepHead, Step)	The first neighbor is the subject of the relation Prep. The second neighbor is the object of the relation Prep.
HasAtt	(Method, Attribute);(Step, Attribute);-(Entity, Attribute);(Attribute, Manner);(MannerHead, Manner)	The second neighbor is the attribute of the first neighbor.

- ***Real*** vs. ***Auxiliary***

The instance of a ***Real*** class is always from the content of a patent claim, while that of a ***Auxiliary*** class is not from the content of a patent claim but can play a role as following:

(a) to link two relations

In a conceptual graph, a relation is not allowed to directly link with another relation. But in reality in a natural language sentence, a relation usually may allow to link with another relation in order to express a complete notion. For example, in the sentence of "*A method for planarizing a wafer, comprising: positioning the wafer on a platen*" from USPTO 7121919, "for" is created as a relation class with label ***Prep*** and "planarizaing", "comprising" and "positioning" are created as relation classes with label ***Verb***. The system creates a Auxiliary concept class ***Action*** to link relation ***Prep***:for to relation ***Verb:***planarizing and to link relation ***Verb:***comprising with relation ***Verb:***positioning as shown in Figure 5.

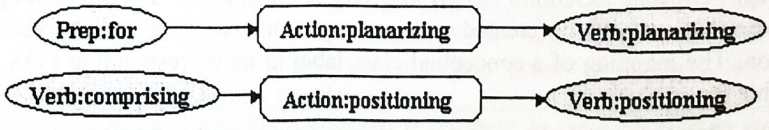


Fig. 5. An example of using a Auxiliary concept to link two relations.

(b) to link two concepts

In a conceptual graph, a concept can not directly link with another concept. The system creates a ***Auxiliary*** relation class to link these two concepts if no ***Real*** relation class can link them. For example, in the phrase of "smectite clay', "smectite" is created as a concept class with label ***Attribute*** and "clay'" is created as a concept class with label ***Entity***. The system uses relation class ***HasAtt***:* to link concept ***Entity***:clay' to concept ***Attribute:***smectite as shown in Figure 6 where ***HasAtt***:* is a ***Auxiliary*** class.

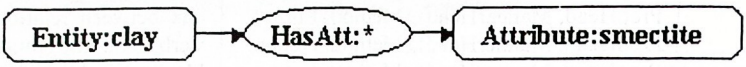


Fig. 6. An example of using a Auxiliary concept to link two concepts.

(c) to link a relation with a set of same concept classes (neighbors)

In a conceptual graph, each relation has a fixed degree (number) of neighbors but it is difficult to decide the degree of neighbors for a relation in advance because a relation may have potentially an infinite number of neighbors. For example, A Verb (Relation) may have more than one Modifier (Concept).

Table 4 shows the types of conceptual class label in terms of either real/Auxiliary

Table 4. The types of conceptual class label in terms of either real/Auxiliary.

Type	Conceptual class label
Real	Method, Step, Manner, Attribute, Entity, Value, Verb, Contains, Prep
Auxiliary	Action, PrepHead, MannerHead, HasAtt

5 Build the Conceptual Graph Based on the Parsed Dependency Trees

To build the conceptual graph from the dependency instances generated by NLP Parser, there are three problems:

Problem 1: *How to decide what the conceptual classes (concepts or relations) of the head and its dependent in a dependency instance and how to assign correct conceptual class label to them?*

For the problem 1, the system divides the Part-Of-Speech into two categories, Concept-POS and Relation-POS. If the Part-Of-Speech of a word belongs to the Concept-POS, it will be created as a concept. Otherwise, it will be created as a relation. The mapping of a conceptual class label to its corresponding POS is shown in Table 5.

Problem 2: *How to link the conceptual classes between a head and its dependent?*

To link the conceptual classes between *a head* and its *dependent*, only three kinds of combinations (concept_1, concept_2), (relation, concept), and (relation_1, relation_2) are allowed. Table 6 shows the dependency relations between a head and its dependent and their corresponding conceptual graph in terms of their dependency

labels, "nsubj", "dobj", "advmod" and "amod". For the dependency label "advmod", because a "verb" may have more than one manner, the system creates a concept "MannerHead" to link the manners of a verb.

Problem 3: *How to combine the conceptual graphs of all sub-sentences?*

To combine all conceptual graphs of each sub-sentence into a single conceptual graph, the system uses the indefinite or definite determiners "A" and "The" respectively to check if a concept is a reference. If a concept C has a definite determiner "The" means that it is a reference, the arc connected to C will be changed to connect to the same concept that has the determiner "A" but has a same name as the concept C.

✍ **A Real Example to extract the concept graph**

We use sub-sentence "*the method comprises a step which introducing the polishing slurry into a centrifuge*" split from US patent 6979252 to show the processes to extract the conceptual graph as shown in Table 2. Figure 7 is the Part-Of-Speech for each word and Table 7 shows the processes of extracting a conceptual graph from dependency tree of the sentence step by step. Figure 8 shows the final conceptual graph of the sentence.

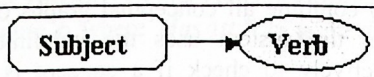
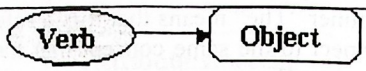
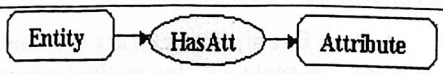
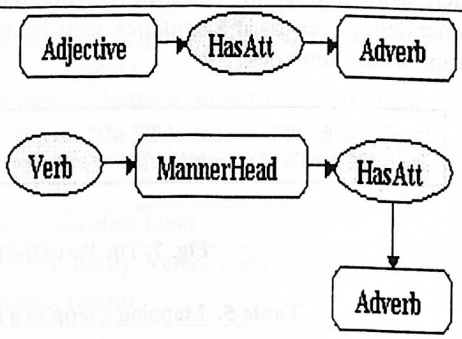
"the/DT method/NN comprises/VBZ a/DT step/NN which/WDT introducing/VBG the/DT polishing/VBG slurry/NN into/IN a/DT centrifuge/NN"

Fig. 7. The Part-Of-Speech of a sentence.

Table 5. Mapping a term to a Real Conceptual Class Label.

Conceptual class	Conceptual Class Label	Part-Of-Speech of the term
Concept-POS	Method	**NN** (Noun, singular or mass) and the term phrase is "method"
	Step	**NN** (Noun, singular or mass) and the term phrase is "step"
	Manner	**RB** (Adverb):**RBR** (Adverb, comparative); **RBS** (Adverb, superlative)
	Attribute	**JJ** (Adjective):**JJR** (Adjective, comparative);**JJS** (Adjective, superlative):**VBG** (Verb, gerund/present participle);**VBN** (Verb, past participle);
	Entity	**NN** (Noun, singular or mass);**NNS** (Noun, plural);**NNP** (Proper noun, singular);**NNPS** (Proper noun, plural);
	Value	**CD** (Cardinal number);
Relation-POS	Verb	**VB** (Verb, base form);**VBD** (Verb, past tense);**VBG** (Verb, gerund/present participle);**VBN** (Verb, past participle);**VBP** (Verb, non-3rd person, singular, present); **VBZ** (Verb, 3rd person, singular, present)
	Contain	The Part-Of-Speech is the same as the Verb relation and the verb phrase must be the transition phrase, "comprising", "having", "including", and "consisting of"
	Prep	**IN** (Preposition);**TO** (infinitival to);**RP** (Particle)

Table 6. The relation between a dependent and its head and their corresponding conceptual graph.

Dependency	The relation between a dependent and its head	Conceptual graph
nsubj	A subject depends on a verb and the conceptual class is **(relation, concept)**	
dobj	A direct object depends on a verb and the conceptual class is **(relation, concept)**	
amod	An adjective depends on a noun, the conceptual class is **(concept_1, concept_2)**. The system creates a ***Auxiliary*** relation "HasAtt" to link the two concepts.	
Advmod	1. An adverb depends on an adjective and the conceptual class is **(concept_1, concept_2)**. The system creates a ***Auxiliary*** relation "HasAtt" to link the two concepts. 2. An adverb depends on a verb and the conceptual class is **(concept, relation)**. The system creates a concept "MannerHead" to link the relation "Verb" and creates a ***Auxiliary*** relation "HasAtt" to link "MannerHead" with the concept "Adverb".	

6 Experiments

The corpus used for our test consists of 1700 patent documents downloaded from USPTO related to the CMP (chemical mechanism polishing) domain by using the search keywords "CMP" and "chemical mechanism polishing" in the Title field. Only the first claim (independent claim) of the 1700 patent documents was selected as the test corpus. From the 1700 patent claims, the average length is 150 words and there are 1458 patent claims whose lengths are greater than 70. Using the split heuristics in Table 3, we obtained 16505 (99%) sub-sentences whose lengths are less than 70.

Table 7. Extract the conceptual graph step by step from dependency tree of a sentnece.

Step	Triple	Extraction of Semantic Relation
1	det(method, The)	Create concept "Method" with generic marker "*"
2	nsubj(comprises, method)	1. Create relation "Contains" with generic marker "comprises" 2. Link "Method:*" to "Contains:comprises"
3	det(step, a)	Create concept "Step" with generic marker "*"
4	dobj(comprises, step)	Link "Contains:comprises" to "Step:*"
5	rel(introducing, which)	Create relation "Verb" with generic marker "introducing"
6	rcmod(step, introducing)	Link "Step:*" to "Verb:introducing"
7	det(slurry, the)	Create concept "Entity" with generic marker "slurry"
8	amod(slurry,polishing)	1. Create concept "Attribute" with generic marker "polishing" 2. Create relation "HasAtt" with generic marker "*" 3. Link "Entity:slyrrt" to "HasAtt:*" 3. Link "HasAtt:*" to "Attribute: polishing"
9	dobj(introducing, slurry)	Link "Verb:introducint" to "Entity:slurry"
10	prep(introducing, into)	1. Create concept "PrepHead" with generic marker "*" 2. Create relation "Prep" with generic marker "into" 3. Link "Verb:introducint" to "PrepHead: *" 4. Link "PrepHead:*" to "Prep:into"
11	det(centrifuge, a)	Create concept "Entity" with generic marker "centrifuge"
12	pobj(into, centrifuge)	Link "Prep:into" to "Entity:centriguge"

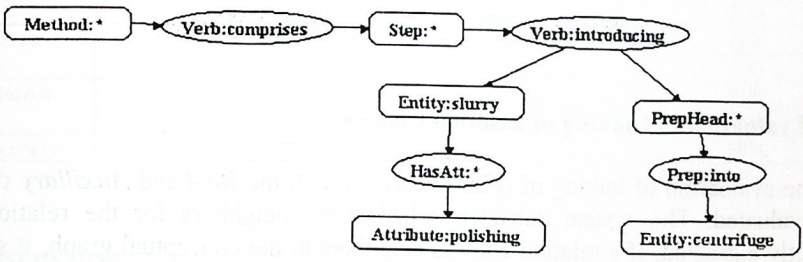


Fig. 8. The conceptual graph of a sentence.

To evaluate the conceptual graph building, we select 100 patent claims as the test corpus and build the correct conceptual graph of each patent claims manually. The system compares the conceptual graphs extracted by the automated approach with those extracted manually and uses precision and recall to evaluate the automated approach.

6.1 Evaluation of Mapping from Contents to Concept or Relation Classes

For the evaluation of mapping from contents to concept or relation classes, only the ***Real*** classes are evaluated because they are extracted directly from the content of the patient claims. The system only consider if it appears in both the content of the original sentence and the corresponding conceptual graph. The measure formulas are below:

$$\textbf{Precision for the class label} = \frac{(\text{PatentClass I CGClass})}{\text{CGClass}} \quad (1)$$

$$\textbf{Recall for the class label} = \frac{\text{PatentClass I CGClass}}{\text{PatentClass}} \quad (2)$$

Where:

- PatentClass : The PatentClass is the set of the concepts or relations in the patent claims.
- CGClass : The CGClass is the set of the concepts or relations in the conceptual graphs.

Table 8 and Table 9 show the precision and recall for the ***Real*** concept and relation classes respectively.

Table 8. The precision and recall for the ***Real*** concept classes.

	Method	Step	Manner	Attribute	Entity	Value	**Average**
Precision	100%	99%	94%	93%	92%	100%	**96%**
Recall	100%	95%	72%	85%	95%	78%	**87%**

Table 9. The precision and recall for the extraction of the ***Real*** relation classes

	Verb	Contains	Prep	**Average**
Precision	96%	98%	98%	**97%**
Recall	95%	99%	98%	**98%**

6.2 Evaluation of Linking of Relation Classes

For the evaluation of linking of relation classes, both the ***Real*** and ***Auxiliary*** classes are evaluated. The system considers whether the neighbors for the relation are correctly extracted. If a relation with its neighbors in the conceptual graph, it should have corresponding neighbors for the corresponding relation in the patent claim.

$$\textbf{Precision for the Neighbor}_i \textbf{ of the relation class } R_k = \frac{\sum_{R_L \in \text{CGRelation}(R_L)} N_\text{Value}(N_i(R_k))}{\text{CGRelation}(R_k)} \quad (3)$$

$$\textbf{Precision for the relation class } R_k = \frac{\sum_{R_k \in CGRelation(R_k)} R_Value(R_k)}{CGRelation(R_k)} \quad (4)$$

Where:

- CGRelation(R_k): The CGRelation(R_k) is the set of the relation R_k in the conceptual graphs.
- $N_i(R_k)$: $N_i(R_k)$ is the i_{th} neighbor of R_k.
- N_Value($N_i(R_k)$) : If the i_{th} neighbor of the R_k in the conceptual graph is the same as the one in the patent claim, then N_Value($N_i(R_k)$) = 1, otherwise N_Value($N_i(R_k)$) = 0.
- R_Value(R_k): $\forall N_i \in R_k$, N_Value ($N_i(R_k)$) = 1 → R_Value(R_k) = 1, otherwise R_Value(R_k) = 0

Table 10 shows the precision of relation extraction for different types of relation classes. It turns out that the most difficult relation extraction belongs to relation class Verb because the average precision of the relation class R_k for Verb is only 70%, while the average precision of the relation class R_k for overall relation classes is about 79%. The last column in Table 10 shows the average precision for each individual neighbor of relation R_k.

Table 10. The precision of relation extraction for different relation classes.

Relation Class R_k	Precision for the relation class R_k	Neighbor$_i$	Precision for Neighbor$_i$ of relation class R_k
Verb	70%	Neighbor$_1$	88%
		Neighbor$_2$	82%
		Neighbor$_3$	90%
		Neighbor$_4$	95%
Contains	77%	Neighbor$_1$	88%
		Neighbor$_2$	87%
Prep	81%	Neighbor$_1$	86%
		Neighbor$_2$	92%
HasAtt	87%	Neighbor$_1$	87%
		Neighbor$_2$	94%
Average	**79%**		**89%**

7 Discussion

The system maps the content in a patent claim to ***Real*** concept class using Part-Of-Speech and the content of the sentence. In the performance evaluation for the concept and relation classes, the recalls of the concept classes for ***Manner*** and ***Attribute*** are much poorer than other concept classes while the recalls for all relation classes (such as ***Verb*** and ***Prep***) are relatively good. It is because that the adjectives and the adverbs are much easier to be tagged wrong than the verbs and the prepositions using the Part-Of-Speech Tagger. In a patent claim, there is a lot of VBG (Verb, gerund/present

participle) such as "deposition" and "comprising" to describe an action. Some of them are erroneously tagged as a Noun that affects the mappings for relation classes ***Verb*** and ***Contains***. Since "comprising" may be followed by "step", if it is erroneously tagged, "step" will also be tagged wrong and therefore it affects the mapping for concept class ***Step***. Some prepositions followed a verb may be erroneously tagged as an adverb that affects the mapping for relation class ***Prep***. There are 36 kinds of Part-Of-Speech generated from the corpus and 20 can be mapped to concepts according to the mapping in Table 7, 7 can not be processed and the rest are special symbols, conjunctions, determiners and punctuations that do need to map to a domain concept.

In the experiment, the precision of neighbor 2 of ***Contains*** is better than that of ***Verb.*** It is because most neighbor 2's of ***Contains*** are ***Step*** while most neighbor 2's of ***Verb*** are ***Entities*** whose precision on concept mapping is worse than those of ***Step***. The neighbor 1 of ***Prep*** is poorer than that of ***Verb*** and ***Contains***, because it is affected not only by the precision of conceptual mapping but also by that of neighbor 3, ***PrepHead***, of the ***Verb*** where ***PrepHead*** is supposed to link all ***Preps*** of the ***Verb***. The precision on neighbor 2 of ***Verb*** and ***Contains*** are heavily affected by the correctness of dependency tree. It is because most objects may have a lot of modifiers before them that cause ambiguities in parsing result. The precision on neighbors of ***HasAtt*** are all much better, because most triples of ***Attribute*** and ***Entity*** or triples of ***Attribute*** and ***Manner*** tend to be correct.

8 Conclusion and Future Work

We have successfully converted a patent claim to a conceptual graph based on the domain ontology. This paper is a pioneer to study on extracting a conceptual graph from a patent claim based on the dependency tree generated from a parser. In the performance evaluation, the average precision and recall for ***Real*** concept class mapping is 96% and 87% respectively and the average precision and recall for ***Real*** relation class mapping is 97% and 98% respectively. It means most terms in the patent claim can be correctly mapped to the domain ontology. The average precision of relation class ***Contains*** is 77% that means most structural information can be extracted correctly. The average precision of relation class ***HasAtt*** is 87% and the average precision of relation class ***Prep*** is 81% that mean the most attributes of an element can be extracted correctly. The average precision of relation class ***Verb*** is only 70% if all the neighbors must be extracted correctly as shown in Table 10 that means most processes or functions in a patent claim cannot be easily extracted correctly. The average precision of each individual neighbor of relation class ***Verb*** is 88% that means it is difficult to improve the precision of extracting relation ***Verb*** by improve the precision of extracting its individual neighbor. After examining the data, we found that most errors were due to the ambiguities generated by the NLP parser. We could design more regular expression patterns to identify more semantic patterns such as the range value ("*solids comprising about 1 to about 10 by weight clay abrasive particles*"), or assignment ("*chemical having a pH greater than 7*") to shorten a patent claim to reduce the ambiguities of applying the parser. In the future,

we will utilize the extracted conceptual graph to facilitate patent processing such as patent summarization and comparison to help the judgment of patent infringement.

References

1. Hobbs, J., Appelt, D., Bear, J., Israel, D., & Tyson, W. M.:FASTUS: A System for Extracting Information from Text, Proceedings, Human Language Technology, USA, (1993) 133-137
2. Sheremetyeva, S.:Natural Language Analysis of Patent Claims. Proceedings of the ACL-2003 workshop on Patent corpus processing Association for Computational Linguistics (2003)
3. Shinmori, A., Okumura, M., Marukawa, Y. & Iwayama, M.:Patent claim processing for readability: Structure analysis and term explanation. In Proceedings of the ACL-03 workshop on patent corpus processing (2003)
4. Sowa, J. F.:Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley (1984)
5. Amati, G. and Ounis, I.:Conceptual Graphs and First Order Logic. The Computer Journal. 43 (2000) 1-12.
6. Marneffe, M., MacCartney, B., & Manning, D. C.: Generating Typed De-pendency Parses from Phrase Structure Parses. Inter-national Conference On Lerc Language Resources and Evaluation (2006)
7. United States Patent and Trademark Office. http://www.uspto.gov/.

A Construction of Emotion Thesaurus Basing on Chinese Character and Empirical Knowledge

Yu Zhang[1], Zhuoming Li[1], Fuji Ren[1] and Shingo Kuroiwa[1]

Faculty and school of Engineering, the University of Tokushima, 2-1 Minamijosanjima-cho,Tokushima 770-8506, Japan
{zhangyu, ren, kuroiwa}@is.tokushima-u.ac.jp

Abstract. This paper presents an effective and practical approach on the emotional lexicographic component of Chinese documentation and metadata specification. In this paper, based on Chinese character and empirical knowledge, we construct an emotion thesaurus in which there are some elements recorded, such as image value, word frequency and affective label which including emotion intensity and affective label and so on. Additionally, a sentence corpus are developed to improve the performance of the emotion thesaurus. Finally, we discuss the advantage of the emotion thesaurus and according to the evaluation experiment this thesaurus produces much better results.

Keywords: Emotion thesaurus (ET), affective label, sentence gatherer, emotion intensity

1 Introduction

Textural information is a huge source of emotion such as words and emotions. Our group is working on emotion extracting from textural information. Emotion can be clearly realized by people from a novel or a story, Researchers first will do lexical analysis then do syntax analysis, and from the keyword we got, we give the emotional information to the keyword using emotion thesaurus we have made.

Why to develop the emotional thesaurus? It is proverbial that Chinese is difficult for non-native speakers, such as using words to make sentences even though they know each word. Because the difficulty is from the descriptive word. When a descriptive word appears in different context it may express different meanings. Every descriptive word have it's own tendency, negative or positive. When using these words people have to know the tendency before using them. Some words are used as politeness and honorifics, while some words are impolite or discriminate and should not be used in friendly commercial correspondence. Along with the international exchanges getting more and more tightly, the request to culture and literalness of all countries also raises, it seems the traditional way is not suitable for the developing condition, by which words are translated only by likeness, and similarity in shape. Although we can't enumerate in detail the civilization for several thousand years of every word, we can consider add the emotional attribute to the vocabularies. That is

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 330–340

Received 12/06/07
Accepted 31/08/07
Final version 21/10/07

considered to be very necessary. Some emotion dictionary have already made in Japanese. [1][2]

For example, "he got praise from his teacher" if do not know the different of "受到/shou dao", "遭到/zao dao" foreigner always make mistakes using "他遭到了老师的表扬" instead of "他受到了老师的表扬". The point on this argument is ignoring the emotional attribute of one word, when the communication between countries become more tightly the understanding of culture and linguistic is becoming important. While in Chinese, regretfully there is no one dictionary containing clear emotional information. Emotion studying and analyzing in Chinese text is still on a starting stage. Lacking of a comprehensive electronic Chinese thesaurus is the crux. There are insistent demands of an emotional thesaurus for more help. Hence, in our study we build up our own corpus and construct an emotional thesaurus. And in this paper we will introduce our work about the construction of an emotion thesaurus (ET).

This paper is organized as follows:

We present the related work and the necessity of emotional thesaurus in our study in section 2. In section 3, we talk about the construction of the emotion thesaurus (ET) detailedly, such as the vocabulary entry, image value, affective label. In the part of affective label we introduce the method of distributing emotion category using Chinese character and empirical knowledge. The section 4 is the evaluation based on the result of investigating experiments. In last section 5 we remark the conclusion and future work.

2 Related Work

Much research has been done on how to classify emotions. In the field of psychology and linguistics, some researchers have done some work with emotion such as emotion classification from facial expression [3], emotion analysis. [4]. In these research emotion are mostly divided into 4-8 kinds. Such as Ekman who had classified emotions into 6 kinds depending on the facial expression, they are happiness, sadness, fear, disgust, anger, surprise considered to be the basic emotion of human beings. To this basic emotion set love was added, since it has been extensively promoted as a basic emotion with the realm of prototype approaches [5]. In some Japanese dictionary [2], there are eight kinds of emotion recorded with enough examples.

In some archaic Chinese literary work, 7 kinds of emotion are often used from a mass of textual recordation. They are "disgust, happiness, anger, sadness, fear, love, and desire". In psychology and common use, emotion is an aspect of a person's mental state of being, normally based in or tied to the person's internal (physical) and external (social) sensory feeling. Some Chinese psychologist have done deep research about emotion, mood of human, they get out their classification basing on point of view of psychology. In contemporary Chinese the emotion word based on psychology and susceptibility can be divided into 24 kinds. [6]

But in our research, the more kinds of emotion don't mean the better the result is, and we must choose the primary emotion as our subject. Comparing the basic emotion

of Ekman's [7], at last we add 6 kinds of emotion depending on the word frequency from 24 kinds and plus (the) equable.

Table 1. 13 kinds of emotion.

Emotion						
Happy Love	Sad Expectant	Fearful Nervous	Disgusted Regretful	Angry Praiseful	Surprised Shy	Equable

3 ET

We have constructed a corpus of emotional words called emotion thesaurus which including about 3700 words selected from two dictionaries and one corpora.p[8][9]

Based on the statistics result in People's Daily tagging corpus of the Institute of Computational Linguistics of Peking University, the word's emotional trends are described and formalized in our dictionary. In order to construct the database of emotional information we select phonetic, part of speech, image value, and the affective label including category of emotion and emotion intensity to compose the items of the thesaurus as shown in Figure 2.

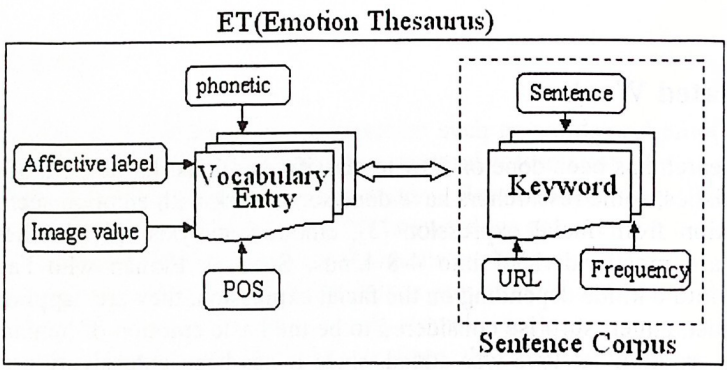


Figure 2. Structure of ET

3.1 Vocabulary Entry

Vocabulary entry is an entry for a word with an emotion-related meaning or connotation. e.g. ,沮丧(gloomy),欣喜(delighted),哭泣(cry)，愿望(desirability), etc. In our ET, we reserve almost all the entries of adjectives, verbs and nouns which denote affects directly. Others like adverbs, some of them with no affect (such as Amplifier words only represent degree) when they are used with some affective words but can strengthen the intensity of words, However, itself has no affective meaning. Words like these, we only give intensity to them. Obviously, POS (part of

speech) is interrelated with every entry and is momentous in emotion judging. There are about 3700 vocabulary entries were gathered in this ET.

3.2 POS

POS is the term used to describe how a particular word is used. All words in the ET are made of nouns, adjectives, verbs, adverbs and also some phrases. From a mass of investigations from Chinese dictionaries [8], the distribution of POS in all of Chinese words is like following.

In modern Chinese, "normal" nouns (i.e. not "time words" or "nouns of locality") make up 53.0% of all vocabulary

Verbs make up 20.1% of all vocabularies

Adjectives make up 4.5% of all vocabularies

Adverbs make up 2.0% of all vocabularies

Phrases make up 8.6% of all vocabularies.

Noun word is used to identify a person, thing, animal, place, and abstract idea of affectively. Adjective word is one of the most important parts of the Chinese Grammar. It is used in describing, identifying a Noun or a Pronoun with affect. A Verb expresses certain actions, events, or states of being. Some emotions are expressed by these words. Not only affective but also some nonaffective words are selected into this ET because the intensity of degree is needed.

那些孩子们欢欣雀跃着，看起来非常开心。----------"非常"（very）

(Those children are gamboling and seem very happy.)

Some words with more than one POS are also gathered as individual entry in the ET. For example:

小心前面的车---------- " 小心 " (beware of) verb. Nervous

他做事很小心---------- " 小心 " (careful) adj. Praiseful

3.3 Image Value

Sometimes a non-native person has trouble in understanding Chinese or misunderstand the meaning of Chinese in listening and speaking. The sticking point is that some words in Chinese have its tendency of + or -, which makes the word own its attitude, such as appraisement, positive and negative. Additionally, for some application of the ET, we find the appraisement of the word is needed to add as the complementary property.

In our ET, there is another intensity of emotion called image value used to scale the plus or minus image of the emotion word.Many authors agree that emotions can be organized roughly into a two-dimensional space whose axes are evaluation (i.e. how positive or negative the emotion is) and activation (i.e. the level of energy a person experiencing the emotion is likely to display)[4]

According to the dictionary of "chang yong baobianyciyu xiangjie cidian", 1103 words are recorded in it, and each word is tagged by the appraisement. This makes our ET more effective. For example, "surprise", in some situation such as, "get a good news", the "surprise" will be positive, while in other situation such as, "I can not believe I fall the exam, it really surprised me" that points to a negative meaning. Worse than negative or better than positive we defined it as derogatory or commendatory and between them we use neutral as a median. Here are the five levels we have defined:

derogatory-> negative-> neutral-> positive-> commendatory

Each word was rated for tendency (positive or negative) and for the degree which presents a given emotion on a five-point scale ranging from –2 to +2 by increments of 1.

3.4 Sentence Corpus

Language corpora usually represent a large collection of representative samples obtained from texts covering different varieties of language used in various domains of linguistic activities. Corpora can be looked as the abbreviation of CORPORA - Capable of Representing Potentially Unlimited Selection of Texts.[10]

The implementation of complex multimedia lexica in hypertext formats is potentially a task of extremely high complexity. It is suggested that as a preliminary step towards designing electronic lexica of various kinds, including Web lexica, a requirements specification in terms of first principles of the lexicon sciences is needed. Electronically-made corpora are new things, and we are yet to reach a consensus to what counts as corpus and how it should be classified.

In this study, we classify the corpora in a very general way, which way focusing on the types of need in the present of Chinese context. They contain written or spoken text, new or old, long or short, which can be obtained from Internet. Obviously the contents are extensive, which can be stories, news, speech, and even they can be extracts of messages of varying length shown in figure 3.

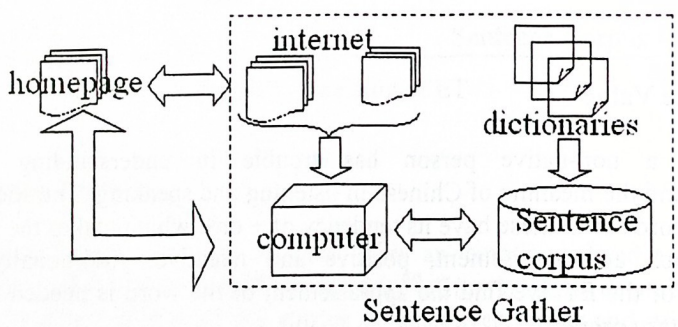


Figure 3. The flow of construct sentence corpus.

In emotion tagging, we consider that one affective word has more than one emotional category, hence the conformation specially includes 13 kinds of emotion which we have defined besides other components. We gather 50 sentences for per

keyword elementarily as the testing text, then distribute the emotion category to the word in one of these 50 sentences.

It is a very tremendous work to give each word tagging of 13 kinds emotion. The development of Internet brings more and more facilities to the people. Everyday there are a fold of textual information gushing from everywhere just like an enormous storehouse of natural language, and it also provides us a lot of usable resources we can depend on. The other advantage of this way is ensuring the functionality of the source. Hence, we exploit a sentence gathering tool to replace gathering sentence from Internet randomly.

3.5 Affective Label

Affective label is a tag of emotion attribute and the emotion intensity. Sometimes the word is not in only one of the 12 categories of emotion, it is in two or three kinds of those emotions. In such an instance every emotion is recorded in our dictionary.

Distributing emotion category. In our study, there are much more detail progresses in the ET. In this ET there are 13 kinds of emotion provided to each word and also the numerical intensity indicated. Such as,

"忧郁"（somber）---adj.---sad 0.36

"忧伤"（somber）---adj.---fearful 0.22

"忧伤"（somber）---adj.---nervous 0.42

Any affect given word may have multiple entries in the ET, differentiated by part of speech and emotion category.

In order to build the ET, there are some preliminary works needed to do such as to indicate the numerical intensity. It is obvious that the emotion has itself ambiguous meanings and sometimes the emotion may puzzle the people. Hence, we use empirical knowledge and method to identify the strength of each emotion to repair the emotion category of words.

Method. Despite its phenomenal salience, (where is the definition of intensity) the intensity of emotion remains a neglected issue of emotion research, and in this study we focus on the intensity of emotions. Emotion intensity represent the strength of affect degree ranging from 0 to 1. In previous publication, there is not a specific criteria of the emotion study, contrarily emotion study is empirical and collect data-sampling depending on the experiment rather than on the elaborate rules.

For a long period in the linguistic study, the empirical data is utilized in lexicography. Samuel Johnson used examples from literature to illustrate his dictionary (1755), while the Oxford English dictionary used citation slips to study and illustrate the usage of words in English. Today, corpora are used for dictionary preparation as they play important role in dictionary building. [11] Moreover, the empirical approach to language study has been identified to be more reliable and authentic than rationalistic (based on intuition) approaches. [12]

In fact, we get the numerical scale of each emotion ascribing to each affective entry from the metadata---sentence. It is a contribution of an empirical database in the form

of corpora. The criteria for scaling emotion intensity is the combination of empirical experiential marking and statistical method.

Here the emotion category of a word can be one or more than one. There are 13 kinds of emotion (shown in Figure 6 and Table 3), which of the word are restricted in these categories. We manually mark the affective word with the emotion category. If there is somewhat any emotional information of the 12 emotions (ex equable), we give the value "1", otherwise the value is "0". Sometimes we can not guarantee that the category is able to be given strictly to the word during the estimation, because the diversified ambiguities inhibit the real emotion. In the Table 3, E13 (equable) is utilized to detract the intensity from other emotions (E1, E2, ..., E12). It means that sometimes the word can be considered as an equable word without any emotion information. In our study, this action is used to get the exact proportion of the emotion.

Table 3. An example of how to mark the emotion. Remark: E1=happy, E2=sad, E3=fearful, E4=disgusted, E5=angry, E6=surprise, E7=expectant, E8=love, E9=nervous, E10=regretful, E11=praiseful, E12=shy, E13=equable

Keyword	No.	Sentence	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13
悲痛	1	东盟 10 国和中国、日本、韩国的外长对以军袭击联合国观察哨所事件感到震惊和悲痛，并向受害者家属表示慰问。	0	1	0	0	1	1	0	0	1	0	0	0	0
悲痛	2	对于刘老师的不幸逝世，我深感十分悲痛，我要化悲痛为力量，接过刘老师高举的伟大红旗。	0	1	0	0	0	1	0	1	0	0	0	0	0
悲痛	..	....	..	..	..	..	..	..	..	..	..	..	..	..	..
悲痛	50	.....	0	1	0	0	0	0	0	0	0	0	0	0	0

We experiment with these 50 sentences in this paper and get the possession of each emotion attribute to these 13 kinds of emotion. Using them as the basic contextual data, we extracted emotion attributes from them at last. (Figure 6).

	13 kinds of emotion	
word	E11 E12 .....E1j..... ... E113	S1
	E21 E22 .....E2j..... ... E213	S2
	....	...
	Ei1 Ei2 ..... .Eij..... .. Ei13	Si
	....	...
	E501 E502 ...E50j..... E5013	S50
Sum(E)	E1 E2 ... . Ej..... ... E13	

S=Sentence

Figure 6. The matrix of the emotion calculation, including 50 sentences and 13 kinds of emotions. Eij (i = 1, ..,50, j = 1, ...,13) is the value "1" or "0" to denote that the word has the emotion information or not. Si (1 = 1, ..., 50) is the number of the sentence.

In these 50 sentences the total of each emotion is $E_j = \sum_{i=1}^{50} Eij$. For each kind of emotion we get a sum, and share it to the 50 sentences, then the average of each

emotion is $Ej_{ave} = \frac{E_j}{50}$. That is considered to be the affective intensity of the 13 emotions of the word. Here Ej_{ave} is a center centage of every emotion distribution in sentences for each word. (The result is shown in the Table 4.)

Table 4. The result of 50 sentences emotion calculation.

Word	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13
悲痛	0.00	0.61	0.01	0.00	0.04	0.11	0.08	0.06	0.01	0.01	0.00	0.00	0.07
开心	0.90	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.06
伤心	0.00	0.51	0.03	0.08	0.03	0.02	0.00	0.11	0.14	0.06	0.00	0.00	0.02
忧愁	0.00	0.37	0.28	0.00	0.00	0.00	0.00	0.00	0.34	0.00	0.00	0.00	0.01
难过	0.00	0.47	0.03	0.09	0.08	0.05	0.02	0.00	0.17	0.01	0.00	0.03	0.04
忧郁	0.00	0.27	0.17	0.07	0.01	0.00	0.00	0.01	0.31	0.00	0.01	0.00	0.16
忠厚	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.50	0.00	0.44
勇敢	0.00	0.00	0.00	0.00	0.00	0.03	0.05	0.08	0.00	0.00	0.41	0.00	0.44
充沛	0.04	0.00	0.00	0.01	0.00	0.00	0.05	0.04	0.00	0.00	0.32	0.00	0.54
...	...	...	...	...	...	...	...	...	...	...	...	...	...
贪婪	0.03	0.03	0.01	0.39	0.14	0.05	0.00	0.08	0.01	0.00	0.00	0.00	0.26

Furthermore, this ET is not only a static dictionary but also a mutual one, we also design a new function to make it active. Provision for its constant growth will reflect the meaning changes occurring in the language. As a result, more and more new words are put into use while some uncommon words are out of use, and some of the words will be recurrently used with the new values. Gradually, over time they will achieve a diachronic dimension representing data obtained from a wider range of time. Thereby we make our sentence corpus grow progressively and the sentences are gathered and keep them updating momentarily. Such corpora will help us to identify new words and phrases, to locate newly coined technical terms, even to know actual date of coming of words to track variation in usage of lexical items and phrases, to observe changes in meanings of words, to follow changes in sentence structures, and to reconstruct the emotion categories of words. At last, the intensity of emotion will be inclined to exactness gradually. The sentence gatherer will automatically update every month. For the new words, we also make a participant IE page to collect informational context of new words, as shown as Figure 5.

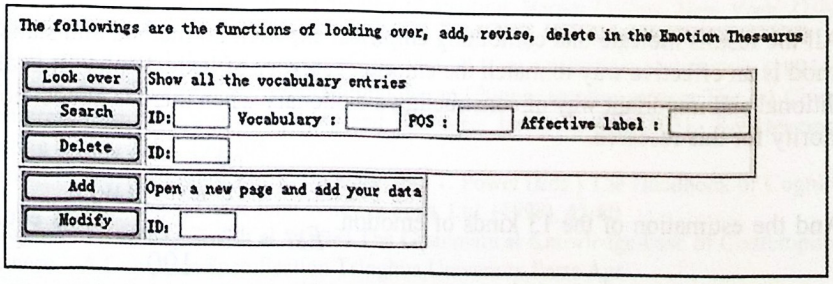


Figure 5. Interface of data updating of ET

4 Evaluation

In order to avoid the category estimation becoming arbitrary and straying somewhat from the strictly affect domain, we invite 14 undergraduate students of the University of Tokushima to participate our experiment, whose native language is Chinese. Each participant is given 100 words with the emotion distributed randomly which are selected from the emotion thesaurus in this paper. The participant is requested to estimate whether the given words with the defined 13 kinds of emotion are compatible to their image or not. If not, the participant will be asked to estimate the correctness of each entry comparing to the expected value. The mark will be a 10-point scale and the decimal digits were limited to 1.

In the evaluation experiment, each word is marked by an emotion value, therefore for each word we get 14 values from all of the participants. We use equation (1) to calculate average evaluation result of 14 participants. Here we remove the max and min marked values, and then the left 12 values are averaged. Figure 7 shows the average evaluation result gotten from 14 participators and 100 words.

$$Est_{ave} = \frac{\sum_{a=1}^{14} Est_a - Est_{\max} - Est_{\min}}{12} \quad (1)$$

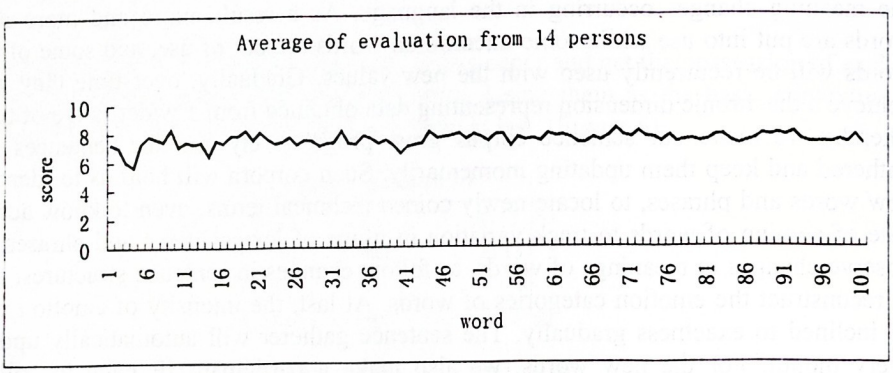


Figure 7. Average of evaluation from 14 persons

All the results indicate that combining empirical experiential method and statistical method is an effective way to match the emotion category of lexical entry. It is also a traditional and important way of constructing a dictionary when there is almost no any authority for this research.

And the estimation of the 13 kinds of emotion $Est = \frac{\sum_{w=1}^{100} Est_{ave}(w)}{100}$ is equal to 7.54 matching to the level D (6-8: somewhat agree). This evaluation result shows that

the emotion value defined in our ET is acceptable and the ET we build in this study is reliable.

5 Conclusion

In this paper, we discussed the current research situations on affective computing according to emotion dictionary and proposed an effective and practical emotion thesaurus based on empirical knowledge and the character of Chinese word to make the emotion definition much more accurate. During the construction of emotion thesaurus, (add something about the content of the thesaurus refer to) we also developed an automatic sentence gathering tool named sentence gatherer to assist to reduce the workload of improving the accuracy of emotion thesaurus. After the automatically sentence collection and manually labeling processing of the emotion words in the selected sentences, the emotion thesaurus construction reconstruction process is terminated. In order to evaluate the ET, we invited some students whose native language is Chinese to evaluate the ET in our evaluation experiment. From the final result, a good correspondence between affect sets and human judgments was discovered. It can be concluded that the emotion estimation in our study is reliable and represents the significant of our study. The ET is proved to play an important role in lexical software application with reusability, interoperability, and portability.

Because the sentence corpus is getting extended with our automatic gathering tool, the ET will be updated momentarily. In future, we will apply it into an emotion classification model as a part of corpus for further research.

Acknowledgments. This research has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), 19300029, 17300065, Exploratory Research, 17656128.

References

1. Ronald Suleski and Masada Hiroko Affective expressions in Japanese : a handbook of value-laden words in everyday Japanese, Tokyo Hokuseido Press, (1982)
2. Yitirou Hiejima : a short dictionary of feelings and emotions in English and Japanese, (1995)
3. Ekman, P., Friesen, W.V., and Elsworth, P.: Emotions in the Human Face, London: Cambridge University Press, (1982)
4. Plutchik, R. : The psychology and biology of emotion. Harper Collins, New York, (1994), p.58
5. Fher, B. and Russell, J.: The concept of love viewed from a prototype perspective. (1991)
6. Xiaoying Xu, Jianhua Tao : Emotion dividing in Chinese emotion system The 1st Chinese Conference on Affective Computing and Intelligent Interaction(ACII'03), 8-9 December, Beijing China, 2003, 199-205
7. Ekman, P., Basic Emotions In T. Dalgleish and T. Power (Eds.) The Handbook of Cognition and Emotion. Sussex, U.K.: John Wiley & Sons, Ltd. (1999), 45-60
8. Shiwen Yu, Xuefeng Zhu et al, (1998) The Grammatical Knowledge-base of Contemporary Chinese – A Complete Specification Tsinghua University Press Apr

9. Xianzhen Guo, Chang Yong (1996) ChangyongBaoBianYiXiangJie dictionary The Commercial Press
10. Dash, Niladri Sekhar and B.B. Chaudhuri: Relevance of Corpus in Language Research and Application, International Journal of Dravidian Linguistics. Vol. 32. No. 2. (2003), 101-122.
11. Mindt, D An Empirical Grammar of the English Verb: Modal Verbs,1995
12. Eric Brill and, Raymond J. Mooney: An overview of empirical natural language processing AI Magazine Contents, Volume 18(4): (1997) Winter, 13-24
13. Bates, J. The Role of Emotion in Believable Agents. Communications of the ACM.37(7), (1994), 122-125
14. Mihalcea, R., and Liu, H. A Corpus-based Approach to. Finding Happiness. In Proc. of the AAAI-CAAW'06, (2006)
15. Zhan Weidong, Chang Baobao, Dui Huiming, Zhang Huarui, Recent Developments in Chinese Corpus Research, The 13th NIJL International Symposium, Language Corpora. Tokyo, Japan. (2006)

Music Composition using the EMILE Grammar Inductor

David Ortega-Pacheco and Hiram Calvo

Centro de Investigación en Computación, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz s/n, esq. Av. Mendizábal, México, D. F., 07738. México

david82d@gmail.com, hcalvo@cic.ipn.mx www.hiramcalvo.com

Abstract: We use the grammar inference engine EMILE for automatic music generation. The input to this algorithm is a collection of texts and then, based in the context of each word, it infers certain grammar rules. This engine has been used mostly to infer natural language grammar rules. In this work, we used the collections of MIDI music files from three classical music composers: Johann Sebastian Bach, Frédéric Chopin and Claude Debussy. We processed each file so that each sentence is formed from the notes found in a musical bar; each word is a musical note with a specific length and volume. We experimented with the capability of creating non-crossing compositions following Pedro P. Cruz-Alcázar, Enrique Vidal-Cruz, 1997. That is, we trained the grammar inductor with the compositions of certain composer, then generated a new composition, and finally we compared it statistically with the original compositions to obtain a degree of similarity. We expected the compositions C, obtained when training with a composer *A* to have a lower coefficient of similarity when they are compared with the corpus of composer *B*. These results are presented in incremental intervals to analyze the impact of the size of the training corpus in the grammar induction process.

Keywords: Music Composition, Grammar Induction, EMILE Grammar Inductor.

1 Introduction

There are several techniques for automatic music composition such as Markov chains [4, 9], fractals [16], chaotic systems [10], cellular automata [3], artificial neural networks [11], grammars [2], etc. In particular, systems for music composition based on grammars have used algorithms such as L-systems [13], and grammar induction algorithms like Sequitur [5].

The goal of grammar induction is to learn in an unsupervised way the syntax of a particular language from a corpus of this language. Grammar induction algorithms are used in several areas, for example computational linguistics, natural language processing, bio-informatics, time series analysis, computer music, etc. Particularly, for automatic music composition and music style recognition some commonly used algorithms are ECGI, K-TSI and ALERGIA [6, 7].

© *A. Gelbukh, Á. Kuri (Eds.)*
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 341–351

Received 24/06/07
Accepted 31/08/07
Final version 30/09/07

In this work we use one of the existing algorithms for grammar induction: EMILE (Entity Modeling Intelligent Learning Engine) [1, 8, 15] for composing music automatically. As far as we know, EMILE has not been used for this purpose.

A particular problem we faced was evaluation—many works which use formal grammars for musical composition generally do not include an objective analysis of the resulting compositions; it is very difficult to tag neutrally if a musical composition is *good* or not [6, 7]. As an immediate workaround, we propose carrying out a statistical test of similarity between each new composition k_i versus each training corpus C_j to obtain a degree of similarity between the new composition k_i and each training corpus C_j. The degree of similarity may define the quality of each new composition k_i, and finally the average of the similarities of various compositions may define the algorithm performance for this application.

2 EMILE Grammar Inductor

EMILE (Entity Modeling Intelligent Learning Engine) is an unsupervised method for grammar induction and it is a method based on categorical grammars. The algorithm tries to obtain the language's grammatical structure using positive samples. The EMILE analysis is based on finding each word's context into each sentence (delimited by a dot (.)) in the training corpus and then carrying out two clustering processes for obtaining a context free grammar which represents the analyzed language (Fig. 1). EMILE can generate new sentences by means of the inducted grammar rules (Fig. 2). [1, 8, 15]

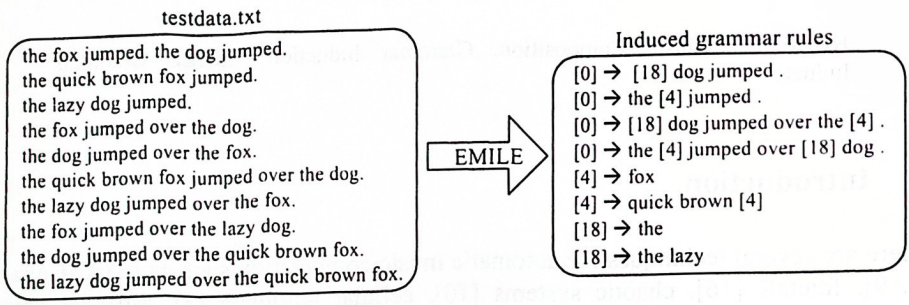


Fig. 1. EMILE grammar induction process [15]

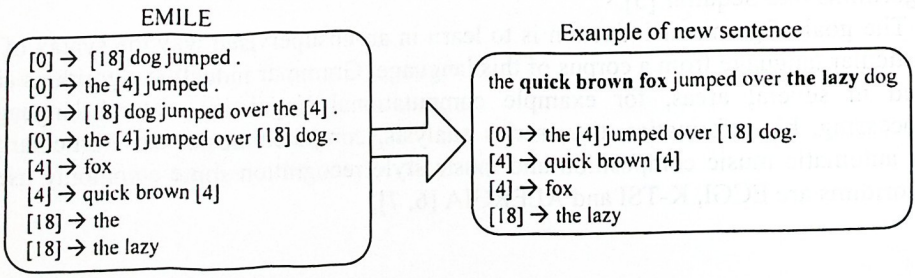


Fig. 2. New sentence generation by EMILE [15]

3 Music Composition using EMILE

The music composition procedure consists of three stages: Obtain the training corpus, Grammar induction and Music composition. We explain each of them below.

3.1 Training corpus

The process to obtain the training corpus for the EMILE grammar inductor is shown in Fig. 3. We explain each part below.

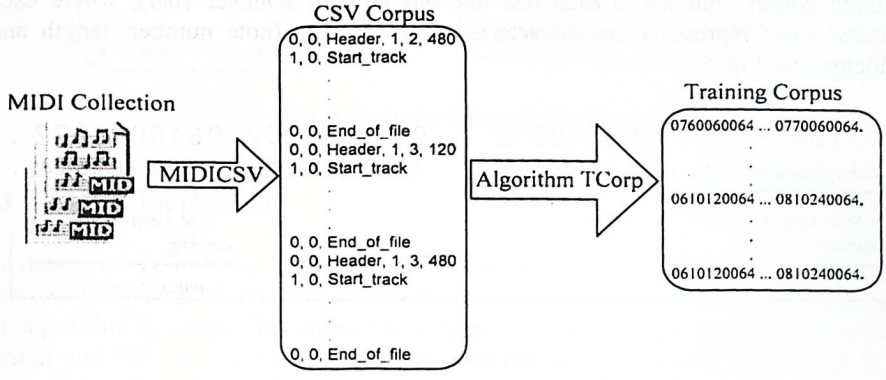


Fig. 3. Process to obtain the training corpus

The musical samples (MIDI Collection) for the Training corpus process are standard MIDI (Musical Instrument Digital Interface) files. We will focus in the events part of a Standard MIDI File [14]: Header, Time_signature, Note_on, Note_off, Note number, Velocity, Time clock, Start_track and End_track.

We convert each MIDI file to a text readable format using the CSVMIDI and MIDICSV programs[1]. This programs, transform a MIDI file to a CSV (Comma-Separated Values) text file and viceversa. CSV files are commonly used to store and transport tabular data between different applications. Fig. 4 shows the process to convert a MIDI file in CSV text format and viceversa.

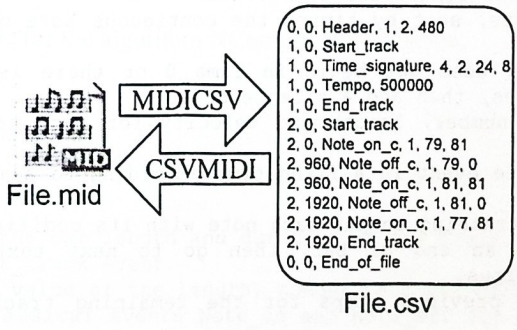


Fig. 4. MIDI file in CSV format.

[1] By John Walker, available from http://www.fourmilab.ch/webtools/midicsv/

We used the MIDICSV tool for obtain the CSV corpus by the MIDI Collection and then extracted the musical events (Header, Time_signature, Note_on, Note_off, Note number, Velocity, Time clock, Start_track and End_track) to create the training corpus for the grammar inductor.

Musical events express the length of each bar, the note number, length and velocity of each note into each bar. We represented each bar in the training corpus with one text line delimited with a dot (.), and each note number and their characteristics of length and velocity with a number of 10 digits, where the first three digits of left to right represent the note number, the next four represent the note length and the next three represent its velocity and each number is separated with one space. Thus, the training corpus contains in each text line one *musical sentence* (bar), where each *musical word* represents the characteristics of one note (note number, length and velocity), see Fig. 5.

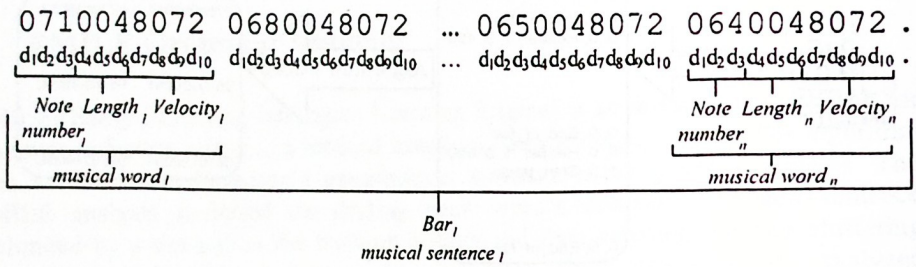


Fig. 5. Format for each text line into the training corpus

We implemented the algorithm TCORP to obtain the length of each bar, the characteristics of each note and the training corpus with the previous scheme for each text line. The pseudo-code for the algorithm is shown below:

```
Algorithm TCorp

Input: CSV Corpus
Output: Training Corpus

Start
  1. Obtain the bar length for each Track.
  2. For each note, sort by timing the contiguous Note_on and Note_off
     events.
  3 If the Track does not begin in Time 0 or there is a time space
     between notes, then add a silence note.
  4. Obtain the number, length and velocity for each note in the CSV
     corpus.
  5. End each note event in a fixed time according to a multiple of the
     bar length.
  6. Add to the training corpus each note with its codified values.
  7. If finding an end of bar, then go to next text line of the
     training corpus.
  8. Repeat the previous steps for the remaining tracks in the CSV
     Corpus.
End
```

3.2 Grammar induction

The input to EMILE is the Training Corpus obtained in the previous process. Once the grammar induction on the training corpus is done by EMILE, the result is a hierarchical structure of composition rules (Fig. 6).

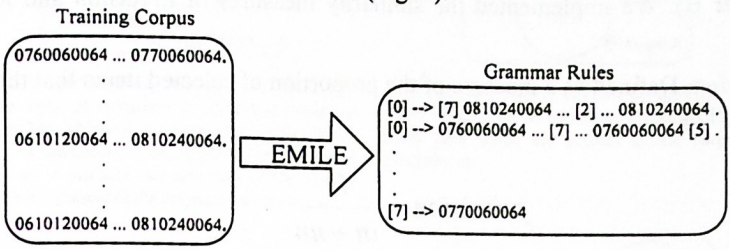


Fig. 6. Grammar induction process

3.3 Music composition

By means of the induced grammar, EMILE can generate new bars which will form a new musical composition. To convert this composition to the MIDI format, we used the algorithm NComp. This algorithm converts the new musical composition to CSV format and then, using the CSVMIDI program we obtain the MIDI file of the new composition (Fig. 7).

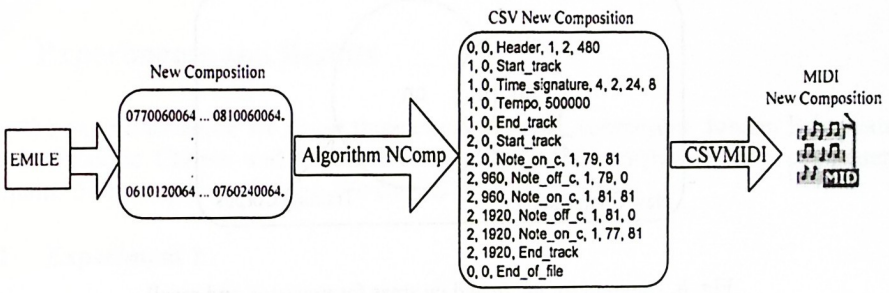


Fig. 7. Music composition process

The pseudo-code for the algorithm NComp is shown below:

```
Algorithm NComp

Input: New Composition
Output: CSV New Composition

Start
  1. Determine the bar length and duration of the Track.
  2. Print Start_track event.
  3. Obtain the value of the length, number, and velocity of each note.
  4. Write the musical events Note_on and Note_off for each note until
     completing the duration of the Track.
  5. Print End_track event.
  6. Print End_of_file event.
End
```

4 Similarity Analysis

The similarity analysis is based on a comparison of each new composition C_i obtained by a corpus A (Composer A) with the same corpus A and with other corpus B (composer B). We implemented the similarity measures of *Precision* and *Recall* as follows:

Precision. Defined as a measure of the proportion of selected items that the system got right [12]:

$$precision = \frac{in}{in + nn} \tag{1}$$

Recall. Defined as a proportion of the target items that the system selected [12]:

$$recall = \frac{in}{in + cn} \tag{2}$$

In the Fig. 8 we show the areas corresponding to new notes (*nn*), intersecting notes (*in*) and corpus notes (*cn*).

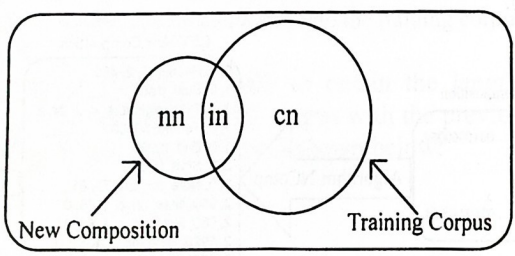
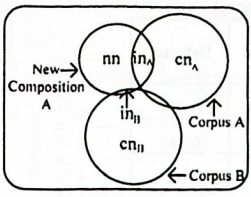
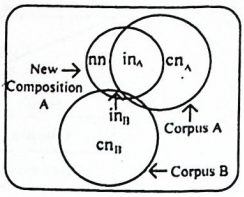


Fig. 8. Diagram for *nn*, *in* and *cn* areas for precision and recall

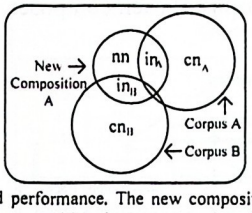
For a good performance of the grammar inductor for this application, we expected that a new composition obtained by a corpus A (Composer A) will have an *Precision A higher than Precision B*, and an *Recall A higher than Recall B*, when is compared with the corpus A and other corpus B (composer B), this may shown that the new composition is an new composition of the composer A. Fig. 9 show samples for a good and bad performance for the grammar inductor.



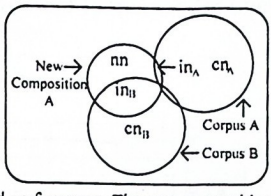
a) Good performance. The new composition A is in the style of composer A (corpus A) because contain bars which are not completely present in the corpus B (composer B) and is a new composition of composer A because contains bars which are not completely present in the corpus of composer A.



b) Bad performance. The new composition is not a new composition of composer A because contains many bars which are present in the corpus of composer A.



c) Bad performance. The new composition is a crossing composition because contains the same number of bars for a composer A and composer B.



d) Bad performance. The new composition A is a crossing composition because is in the style of composer B (corpus B) because contain bars which are not completely present in the corpus A (composer A).

Fig. 9. Good and bad performance for the grammar inductor

5 Experiments and Results

For the experimentation we chose three classical music composers: Johann Sebastian Bach, Frédéric Chopin and Claude Debussy. The MIDI corpus of each composer contains 50 monophonic Tracks. We carried out 2 experiments.

5.1 Experiment 1

1. Generate for each composer his corresponding training corpus.
2. Obtain for each composer his grammar rules (rules for composition).
3. Create 10 new compositions for each composer.
4. Carry out the similarity analysis between each new composition obtained and the corpus for each composer.
5. Repeat the previous steps for each training corpus of each classical music composer.

Table 1. Average Precision.

Corpus / Compositions	Bach	Chopin	Debussy
Bach	**91.476**	20.928	19.621
Chopin	21.682	**89.873**	44.822
Debussy	33.034	38.137	**93.977**

Table 2. Average Recall

Corpus / Compositions	Bach	Chopin	Debussy
Bach	**3.488**	0.059	0.057
Chopin	0.1	**0.177**	0.055
Debussy	0.179	0.11	**0.44**

5.2 Experiment 2

Consisted on carrying out experiment 1 varying the size of the training corpus using 20, 40, 60 and 80 percent of the original training corpus of each composer.

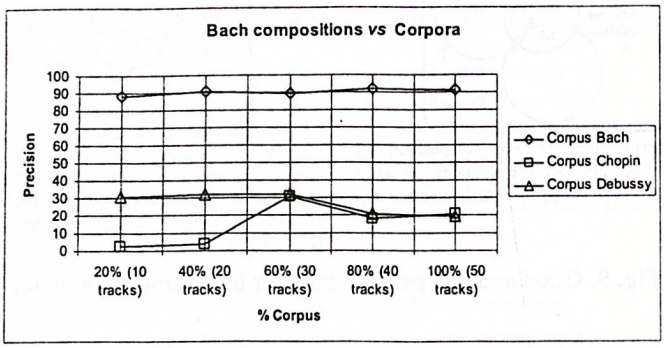


Fig. 10. Precision obtained by varying the size of the training corpus of Bach

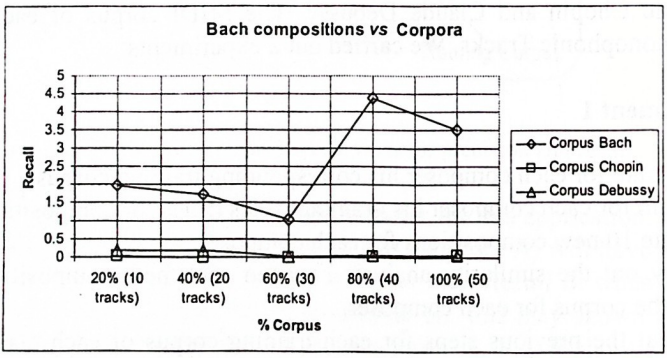


Fig. 11. Recall obtained by varying the size of the training corpus of Bach

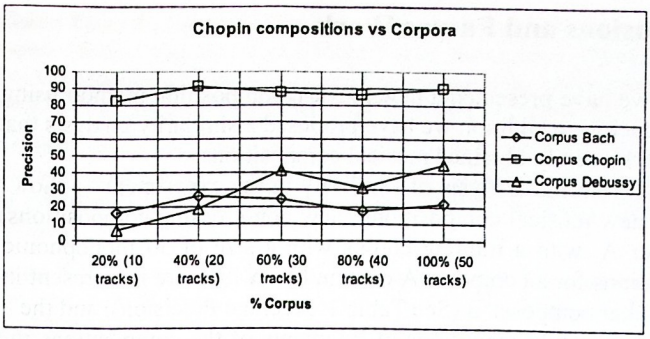


Fig. 12. Precision obtained by varying the size of the training corpus of Chopin.

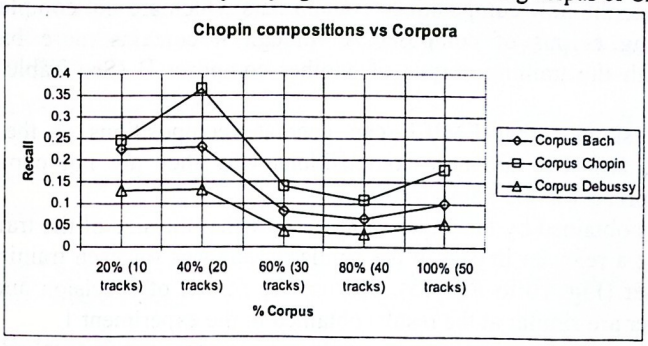


Fig. 13. Recall obtained by varying the size of the training corpus of Chopin

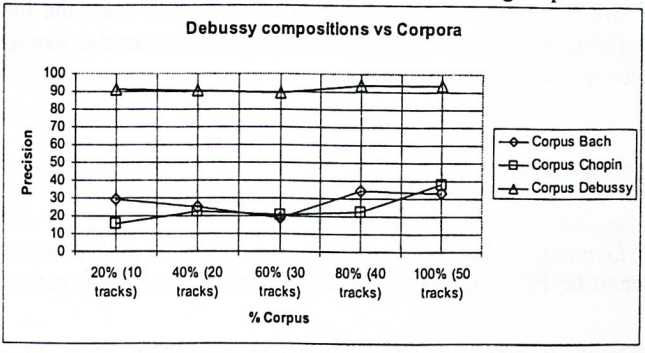


Fig. 14. Precision obtained by varying the size of the training corpus of Debussy

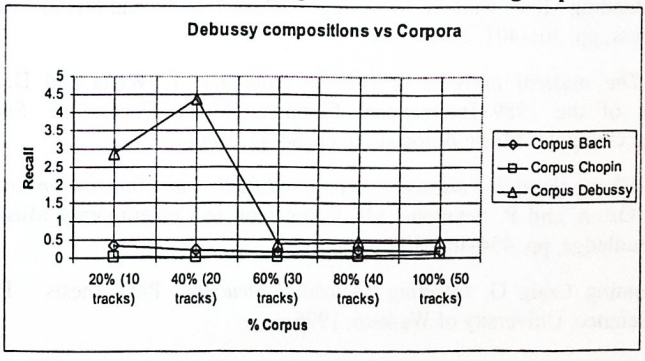


Fig. 15. Recall obtained by varying the size of the training corpus of Debussy

6 Conclusions and Future Work

In this work we have presented a new musical composition method using the EMILE Grammar Inductor; in addition we have proposed a similarity analysis that may define the quality of the newly obtained musical compositions.

The results from the analysis of similarity in the experiment 1, show that EMILE can generate new musical compositions (new non-crossing compositions) in the style of a composer A, with a training corpus with a size of 50 monophonic tracks. The new compositions for a composer A contain bars which are not present in the training corpus of another composer B (See Table 1. Average Precision), and the compositions obtained are not merely repetitions of fragments of the compositions in the learning corpus because the new compositions contain bars which are not completely present in the training corpus of composer A—though it contains more bars when is compared with the training corpus of another composer B (See Table 2. Average Recall).

This may suggest that EMILE can generate compositions in the style of a composer A, and they are new compositions why they are not copies of some composition present in the training corpus.

Our results obtained by the experiment 2, show that the size of the training corpus does not have a relevant impact in the similarity analysis for each training corpus of each composer (Fig. 10 to Fig. 15), because the results of Precision and Recall for each composer are similar at the results obtained in the experiment 1.

As a future work, we plan to compare this music composition method using other algorithms of Grammar Induction, different scheme for represent the notes and their characteristics (Note number, Length and Velocity), increment the size of the training corpus and incorporate polyphonic tracks.

References

[1] Adriaans P. *Learning Shallow Context-Free Languages under Simple Distributions*. ILLC Research Report PP-1999-13, Institute for Logic, Language and Computation. Netherlands, 1999

[2] Bel B., J. Kippen. *Bol Processor Grammars*. In M. Balaban, K. Ebcioglu, and O. Laske, eds. Understanding Music with AI, Cambridge , Massachusetts and Menlo Park, California: The MIT Press, pp. 366-401, 1992

[3] Beyls P. *The musical universe of cellular automata*. T. Wells and D. Butler, eds. Proceedings of the 1989 International Computer Music Conference. San Francisco: International Computer Music Association, pp. 34-41, 1989

[4] Chadabe J. *Flying through a musical space: about Real Time Composition*. In J. Painter, T. Howell, R. Orton, and P. Seymour, eds. Companion in Contemporary Musical Thought. London: Roudledge, pp. 454-465, 1992

[5] Nevill Manning Craig G. *Inferring Sequential Structure*. PhD Thesis - Philosophy in Computer Science, University of Waikato, 1996

[6] Cruz Alcázar Pedro P., Enrique Vidal Ruiz. *A Study of Grammatical Inference Algorithms in Automatic Music Composition and Musical Style Recognition*. Workshop on Automata Induction, Grammatical Inference, and Language Acquisition. The Fourteenth International Conference on Machine Learning (ICML-97), Nashville, Tennessee, USA, 1997

[7] Cruz Alcázar Pedro P., Enrique Vidal Ruiz. *Learning Regular Grammars to Model Musical Style: Comparing Different Coding Schemes*. Grammatical Inference, Lecture Notes in Artificial Intelligence, 4th International Colloquium, ICGI-98, Ames, Iowa, USA, pp. 211-222. July 1998

[8] Dörnenburg Eric. *Extension of the EMILE algorithm for inductive learning of context-free grammars for natural languages*. Master's Thesis, University of Dortmund, United Kingdom, 1997

[9] Frankel Goldwater Lee. *Computers Composing Music: An Artistic Utilization of Hidden Markov Models for Music Composition*. Department of Computer Science, University of Rochester, 2005.

[10] Gogins M. *Iterated Function Systems Music*. Computer Music Journal, 15 (1): 40-48, 1991.

[11] Griffth N., Peter M. Todd. *Musical Networks, Parallel Distributed Perception and Performance*. The MIT Press, England, pp. 227-260 1999.

[12] Manning Christhopher D., Shütze Hinrich. *Foundations of Statistical Natural Language Processing*. Second Printing. England, MIT Press, 2000

[13] Prusinkiewicz P. *Score generation with L-systems*. Proceedings of the International Computer Music Conference '86, pp. 455–457, 1986.

[14] Selfridge Field Eleanor. *Beyond MIDI, the Handbook of Musical Codes*. The MTI Press, Cambridge, Massachusetts, London, England, 1997

[15] Vervoorte M. *Emile 4.1.7 User Guide*. University of Amsterdam, 2004

[16] Waschka R., A. Kurepa. *Using fractals in timbre construction: an exploratory study*. In T. Wells and D. Butler, eds. Proceedings of the 1989 International Computer Music Conference. San Francisco: International Computer Music Association, pp 159-161, 1989

Managing the workflow of massive feeding of digital libraries *

Ángeles S. Places, Nieves R. Brisaboa, José R. Parama,
Oscar Pedreira, Diego Seco

Database Laboratory, Facultade de Informatica, University of A Coruña
Campus de Elviña s/n, 15071 A Coruña, Spain
{asplaces,brisaboa,parama,opedreira,dseco}@udc.es

Abstract. Feeding a digital library requires a significant effort due to the cost of a massive digitization and processing of documents, specially when these documents are not in a good state of conservation. When a big group of people work together in this complex process, support tools are needed to manage the process workflow, to facilitate the work in each activity and thus, to reduce the cost and ensure the quality of the obtained results. This paper proposes a set of strategies to face the management of the workflow of the digital library building process and a general system architecture supporting them. The paper also presents a tool called DigiFlow developed following this architecture, and the results obtained from a real experience applying it in the digitization of a collection of 23,000 documents of the 19^{th} century.

Key words: Digital library, digitization, workflow management

1 Introduction

The interest and research on digital libraries has experienced an important growth during the last years, in parallel with the advances in document digitization, information retrieval and web publishing technologies. Among all the applications of these type of systems, one of the most important is the development of digital libraries that store and provide access to collections of ancient documents. The digitization, storage and indexing of these type of documents guarantees the correct preservation [1] of old texts that will not be reedited and that are in a critical state of conservation. In addition, electronic editions provide public access to a collection of documents that would be impossible otherwise [2]. The special problems that appear in this type of systems have opened interesting challenges in the development of digital libraries.

* This work has been partially supported by "Xunta de Galicia" grants PGIDIT05SIN10502PR and 2006/4, and by "Ministerio de Educación y Ciencia" (PGE y FEDER) grant TIN2006-15071-C03-03, and (Programa FPU) AP-2006-03214 (for O. Pedreira)

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 352–362

Received 09/06/07
Accepted 31/08/07
Final version 30/09/07

Perhaps the main problem in digital libraries storing ancient documents is the importance and complexity of the digitization and processing of documents. These documents are usually very old and their state of conservation is, in general, poor. The characteristics of the documents can make necessary the use of special hardware for scanning. In addition, this activity has to be done with care to avoid new damages to the document. The conversion of the obtained images to text through character recognition technologies is specially difficult, due to the deterioration of the documents. Thus, the results of this task must be reviewed in order to correct possible errors. Other activities as metadata definition, document markup and text indexing are usually needed. Sometimes this process is carried out step by step by a small group of experts, and their skills and knowledge guarantee the quality of the results. But when the digital library has to be built from thousands of documents, the creation of the document repository involves a large digitization process carried out by a big group of people who are not always so skilled due to financial restrictions [3]. In these situations, the use of support tools to guide the workflow of the process and to facilitate the work of the people is mandatory.

The document repository building requires the ordered execution of a set of activities on the documents, with several people participating in each of them. In such a complex process, the lack of control on the workflow can result in dead times, errors in the obtained results and loss of data. In general, an unsatisfactory coordination of the people increases the overall cost and decreases the quality of the results. From previous experiences, we identified typical errors like the digitization of the same document several times with different names, errors according file naming conventions, publishing of no reviewed documents, loss of files, missing documents, etc. Because building a digital library requires significant effort, the more automated tools that can be built and used, the better will be the use of human resources [4]. The control of the workflow inside this work team is a key factor in the success of the digitization process. This control can be achieved by the use of a workflow management tool specially designed for this process. That is, a system which allows to coordinate and control all the involved people, monitor and manage factors as the current state of each page, store intermediate results, control the average time to process each document, record all the people who have worked in each document, etc. Previous works have developed support tools for the creation of digital libraries, but they do not consider the problems found in the digitization and process of a large collection of documents [5, 6].

This paper proposes a set of strategies for the workflow management of the repository creation process, and a general system architecture supporting them. The proposed strategies improve the performance of the process, ensure that all the necessary tasks are correctly performed and facilitate the work of the people devoted to this activity. In addition, we have implemented a tool called DigiFlow following this architecture. In this paper we also present the real experience of building a digital library for the *Royal Galician Academy*, from a collection of

23,000 documents of the 19^{th} century. We also present quantitative data about the results obtained in each step of the document repository building.

The rest of the paper is organized as follows. Next section analyzes the main problems found in the massive digitization and processing of documents for large digital libraries. Section 3 presents the requirements and the architecture to solve these problems. Section 4 presents the general architecture for workflow management in the digitization process. Sections 5 and 6 describe the implemented tool and the results obtained in its application to a real project. Finally, Section 7 presents our conclusions and directions of future work.

2 Problems in massive digitization of documents

One of the main problems found when building digital libraries is the complexity and cost of the digitization and processing of documents. Each page of a document has to be scanned, processed using OCR technologies for text extraction, corrected and reviewed. Other activities as metadata definition or document markup are also necessary. Finally the text is indexed, stored and published. Taking into account that the collection of documents can have hundreds of thousands of pages, the complexity and cost of document digitization is obvious. From our previous experience in the development of digital libraries, we have identified several typical problems in this process:

- Problems with the file naming protocol. Due to the high number of files to be managed during the digitization, such a protocol is necessary. When few people participate in the digitization, the file naming conventions are usually followed, and small errors can be easily managed. However, when tens of people are working together, small errors are likely to appear, and their management can produce a significant waste of time.

- Loss of files. Without support tools, each participant is the responsible of the files obtained in each activity. If the management of hundreds or thousands of files is done manually, typical errors will happen frequently. Overwriting files, saving files with the wrong name or in the wrong folder are a common source of lost files. If the experience of the participants with computers is limited, these errors will be very common.

- Task specification. There are different ways to carry out a task. A bad specification of the task parameters is also a source of typical problems. For example, scanning with an incorrect orientation of the pages, scanning two pages together instead of one, reviewing an already reviewed document or writing again the document metadata when they were already available in the database. This problem can be worst when several people work with the same document.

- Lack of coordination. Coordination is difficult when a big group of people work in the project. Each person can be devoted to specific activities and

have his/her own timetable. For example, a given person can be the responsible of scanning a document in the morning and other can be the responsible of correcting it in the afternoon. An effective task management to avoid dead times and waste of resources is necessary.

- Effective resource control. Since the number of resources used for the digitization is limited, the lack of control of them can be a source of dead times in some activities of the chain. For example, some workers could have to wait for free scanners or computers, or even for the availability of the physical document. In addition, without this resource management, reports about the particular resource used in each activity are not available.

- Responsibilities management. The correct definition of the responsible of each task is also important, specially when the review of the extracted texts is difficult and requires deep knowledge of the type of literature being digitized.

Perhaps most of these problems seem trivial and easy to solve. But taking into account that they can be repeated thousands of times during the whole digitization process, their consequences can have a really great impact in factors such as the process time and the quality of the digitized documents.

3 Requirements for the workflow management system

As a solution for the problems presented in the previous section, we describe here the workflow strategies we suggest for the workflow management architecture we propose in this paper.

- Automatized results management. As we mentioned in previous section, errors according the file naming conventions and loss of files are common in a group of people working together in a digitization chain. A workflow management system for document digitization should automatically manage the files produced in each activity. Thus, when an worker starts a new task, the system must automatically bring him/her the inputs needed for such a task (which are outputs of previous tasks), without any kind of human intervention. This support avoids problems of lost of intermediate products.

- Task specification. The system should offer to the administrator tools to continuously monitor different aspects of the workflow. Particularly the state of each document digitization task, which was assigned to a worker is of special interest, but also the progress of the results and possible problems recorded.

- Task control and monitoring. The system should offer to the administrator the tools needed to continuously monitor all the information about the state of each document task assigned to each worker. For example, the worker assigned to the task, the progress of the results and possible problems recorded.

- Effective resource management. This requirement is related with the previous one. The system should continuously control the availability of the necessary resources for each activity, identifying and informing immediately about possible conflicts between tasks due to the used resources. For example, if several documents are being scanned at the same time and a rescanning is needed to correct OCR errors, the system should identify a time slot in which the hardware will be available.

- Work dedication reporting. It is important to provide the possibility of generating reports about the average time devoted to each task, the average number of pages in a period of time, the number of corrections of the results of the OCR, the average dedication of each worker in a given period of time, etc.

- Product quality control. As we mentioned previously, when dealing with ancient documents, the review of the results obtained from the digitization can be really important due to the high error rate in the results of the OCR. Thus, the system must facilitate this review process, providing the reviewer with both the image and the extracted text without the need of obtaining the original document, and ensuring that the document is not published until the review is successfully finished.

4 System architecture

According to [7], workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants following a defined set of rules to achieve, or contribute to an overall business goal; the computerized facilitation or automation of a business process, in whole or part. Workflow management systems can be classified in several types depending on the nature and characteristics of the process [8, 9]. Collaborative workflow systems automatize business processes where a group of people participate to achieve a common goal. This type of business processes involves a chain of activities where the documents, which hold the information, are processed and transformed until that goal is achieved. As the problematic of document repository building fits perfectly in this model, we based the architecture of the system in this model.

In general, we can differentiate three user profiles involved in the repository building:

- Administrator. Administrators are the persons responsible of the digitization process as a whole. They are the responsible of assigning tasks to different workers and controlling the state of each digitized document.

- Advanced users. The advanced users are the persons in charge of carrying out critical activities such as the metadata storage or the review of the texts

extracted from the OCR processes. The rationale behind this user type is that they usually need deep knowledge of the documents for performing these tasks (for example, a deep knowledge of the *Galician* literature of the 19^{th} century is needed if the user is going to review this type of documents).

- Standard users. The standard users are the workers who carry out tasks as scan or the OCR correction. This role is played by users with some knowledge in the document field but without any responsibility on the management of the system (for example, granted students could carry out these activities).

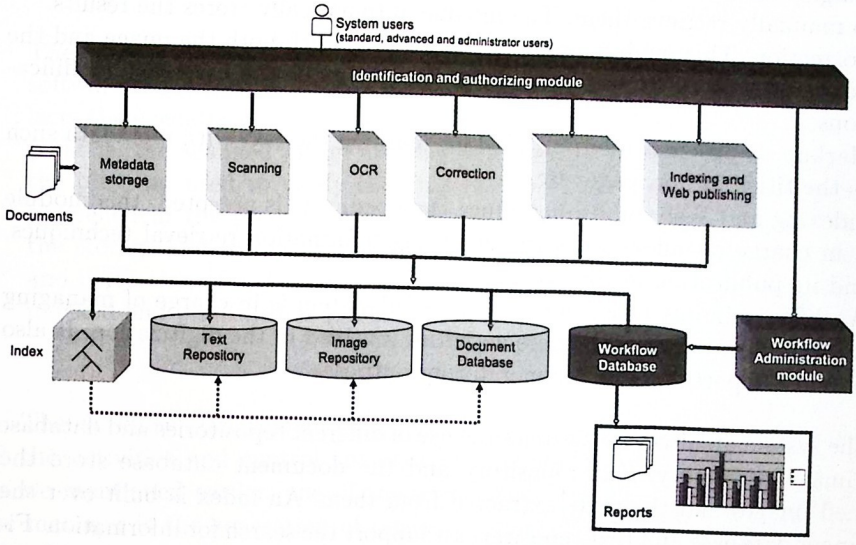


Fig. 1. System architecture

Figure 1 shows the overall system architecture. When defining it, we followed the recommendations of the Workflow Reference Model [8], a commonly accepted framework for the design and development of workflow management systems. Intended to accommodate the variety of implementation techniques and operational environments which characterize this technology. Thus, although we used this architecture for the implementation of a specific system, it can be used in other environments and situations.

As we can see in Figure 1, the authentication and authorizing module is in charge of the authentication of the workers who want to access to the system. Each user has a system role depending on the tasks he/she is going to work on. In terms of this system role, the authorizing module only provides the user with access to the needed features. The system architecture is composed of a module for each activity carried out during the repository creation.

- Metadata storage. This subsystem is in charge of the introduction and storage of the metadata for each document (title, author, year, source, etc.), following any desired format, as Dublin Core [10] or MARK [11]. This task is performed by the advanced users of the system, therefore only them have access to this module.
- Scanning. This system provides access to the scanning hardware and software, and it is the responsible of managing the specification of the scanning parameters for each document (for example, options like scanning two pages at the same time, landscape orientation, resolution, number of colors, etc.).
- OCR. It provides access to the OCR software, and obtains the scanned images needed as the input of this activity, therefore it is not necessary to manually retrieve them. The module automatically stores the results.
- Correction. This module provides the reviewer with both the image and the extracted text to carry out the correction to make the necessary modifications.
- Markup. It provides the tools used for marking the text with metadata such as the title, author, page, etc.
- Indexing and Web publishing. Once the document is accepted, this module is in charge of indexing its content using information retrieval techniques, and its publication in the Web.
- Workflow administration module. This subsystem is in charge of managing the workflow between all these activities involved in the digitization. It also provides reporting tools for monitoring purposes.

The system architecture assumes the use of different repositories and databases. The image repository, text repository and the document database store the scanned images and the texts extracted from them. An index is built over the document database and text repository to support the search for information. Finally, the workflow database stores the information about the digitization chain, with the lists of tasks, the state of each document, etc.

5 DigiFlow: A Tool for Document Repository Building

The architecture presented in the previous section was applied in the implementation of DigiFlow, a workflow management system supporting the digitization process of digital libraries. This system was used in the case study presented in next section. Thus, in this section we briefly describe some issues regarding to the system implementation.

DigiFlow provides an integrated environment where all the tasks can be executed. The idea is that this application provides the user with all the tools needed to carry out each task, without being necessary to use other software applications or manually manage the results of each task. Once the user credentials are introduced and validated in the system, the application shows the list of pending tasks. Once he/she selects the execution of any of them, the system

retrieves the necessary resources and opens the corresponding applications. For example:

- When the user selects a scan activity, the system shows the specification of the task to the user, indicating the document identifier, the page numbers to scan and the scanner options. All this information is shown within the same application. When the user presses the "Scan" button, the document is scanned with the specified options and once the user confirms the validity of the obtained result, the system stores the resulting image and creates the corresponding OCR task. Notice that the user does not have to manually manage any of this products.

- The same approach is followed by the execution of the correction task. The software used for this activity is OmniPage Pro 12 [12]. When the user selects the execution of this task, DigiFlow retrieves the inputs for the review (that is, the scanned image and the text extracted with OCR) and opens the program used to verify that the text corresponds with the image. As happened in the previous example, the user does not have to care about the storage and retrieval of the intermediate documents. When the review and correction of a document is completed, the document is automatically indexed and published in the Web of the digital library.

The workflow management subsystem provides all the tools needed to define all the activities and control the results obtained in each of those activities. As we mentioned earlier, one of the most useful tools the system provides to the manager is the generation of reports about the digitization process, which allow him/her to monitor and control the progress of the project and to identify possible problems. DigiFlow offers three types of reports to the administrator:

- Opened document report. This report informs the administrator of the current state of each document, the tasks to be done and the results obtained in each of them.

- Activity report. This report shows the administrator statistics about each activity of the digitization process. This allows the administrator to have an idea of the effort devoted to each activity and to identify possible bottle necks.

- Users report. This report summarizes the dedication of each worker to each task, such as the work load for each of them and the degree of completion of the assigned tasks. This allows the administrator to assign new activities to the free workers.

6 Case Study: A Digital Library for Ancient Documents

In this section we present the results obtained in the application of DigiFlow to build a digital library of an institution responsible of the care of an important collection of old documents.

6.1 The institution - *Royal Galician Academy*

Royal Galician Academy (RAG) is an institution devoted to the conservation and promotion of the cultural heritage and the language of Galicia, a region located in the northwest of Spain [13]. One of the main characteristics of this region is its rich culture, its own language, and a rich literature. RAG is the responsible of the maintenance and conservation of an archive that stores a big collection of old documents. As an initiative for the promotion of this collection, this institution decided to build a digital library which is compounded of journals of the 19^{th} century. These journals are a patrimony of great value, being many of them unique and showing the situation of Galicia at that time. The publication of this documents in a digital library facilitates their conservation and makes them available to any interested researcher and to the public.

The whole collection contains 23,000 documents. Due to their antiquity and their state of conservation, the digitization of this documents was really difficult. An example of these documents is shown in Figure 2, and all the digitized documents can be accessed in [14]. The size of the collection and the deep knowledge needed to correct the mistakes that appear in the digitization also contributed to the difficulty in building the document repository.

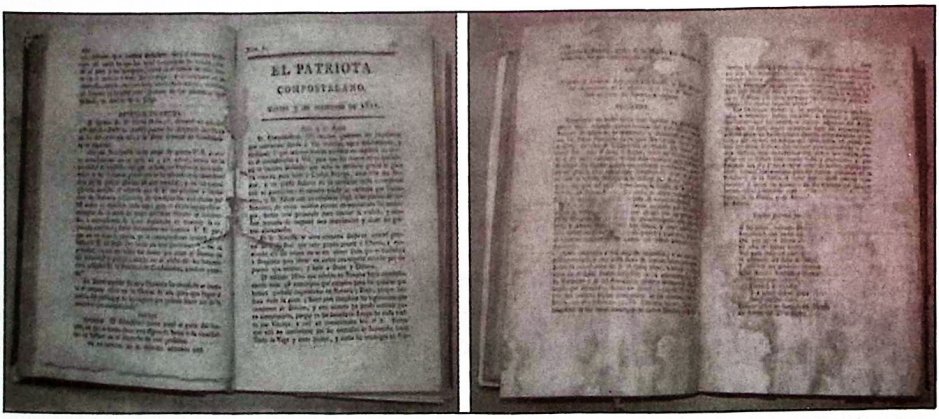


Fig. 2. Images from "El Patriota Compostelano" (1810)

6.2 Building the digital library

The building of the document repository was carried out by a collaboration between the Databases Laboratory (our research group) and the Laboratory of Linguistic Technology of University of A Coruña. A group of twenty granted students from the Laboratory of Linguistic Technology worked in the digitization of documents. Due to their deep knowledge of Spanish and Galician language and of this kind of documents, they are skilled enough to successfully carry out this work.

The hours spent in each one of the main workflow activities are shown in Table 1, as well as the number of pages processed in that period. This information can be used to monitor and take under control such a costly process, and plays an important role as estimation data for future projects of the same characteristics that are planned to be executed. For example, in this information we can see quantitatively the high cost of the correction tasks in comparison with the others when working with old documents in a bad state of conservation.

ACTIVITIES	PAGES	HOURS	PAGES/HOUR
Metadata storage	23,000	235.99	97.46
Scanning	23,000	442.17	52.02
OCR	22,635	658.37	34.38
Correction	21,077	7,611.49	2.77

Table 1. State of the work.

7 Conclusions and future work

The creation of a document repository is not a simple process. It requires the coordination of people and tools to carry out every activity that is part of the process. These activities include digitization of documents, optical character recognition, results correction and indexing to perform search by content. For all these process to be correctly and efficiently made, it is necessary the use of support tools that facilitate the work of each participant and ensure the quality of the obtained results.

The proposed workflow strategies and system architecture support the control and coordination of people and tasks involved in the digitization process. The use of this architecture automates the completion of prone to error activities and optimizes the performance of the digitization process and the quality of the obtained results. This architecture was applied to the design and development of DigiFlow, a collaborative workflow management system designed to create document repositories. This system was built as a desktop application which provides an integrated environment for the execution of all the tasks.

DigiFlow was successfully applied in the building to a digital library for "Royal Galician Academy", an institution devoted to the conservation and promotion of the Galician culture and language. This digital library involved the

digitization of a set of 23,000 documents of the 19^{th} century in a bad state of conservation. In this project, the proposed architecture showed its benefits automatizing all the digitization chain. In this paper, we present this experience with quantitative results of the effort devoted to each task of the project. Although the system was used in a specific case, the proposed architecture is general enough to be applied in any project of these characteristics.

Our work line still maintains opened some questions that will be addressed in the future. First, the developed system is going to be applied to new projects of similar characteristics involving the digitization of hundreds of thousands of pages. In addition, we are working on different implementations of the activities considered to optimize the performance of the overall process. We are also working to provide the system with new compression and indexing tools necessary for really big digital libraries.

References

1. Ross, S., Hedstrom, M.: Preservation research and sustainable digital libraries. International Journal on Digital Libraries **5**(4) (2005) 317–324
2. Borgman, C.L.: Challenges in building digital libraries for the 21st century. In: 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, December 11-14, 2002. Proceedings. (2002) 1–13
3. Chang, N., Hopkinson, A.: Reskilling staff for digital libraries. In: Digital Libraries: Achievements, Challenges and Opportunities. Volume 4312 of Lecture Notes in Computer Science., Springer (2006) 531–532
4. McCray, A.T., Gallagher, M.E.: Principles for digital library development. Communications of the ACM 44(4) (2001) 49–54
5. Witten, I.H., Bainbridge, D., Boddie, S.J.: Power to the people: End-user building of digital library collections. In: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries JCDL '01. (2001) 94 – 103
6. Bainbridge, D., Thompson, J., Witten, I.H.: Assembling and enriching digital library collections. In: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries. (2003) 323 – 334
7. Hollingsworth, D.: Workflow management coalition - the workflow reference model. Technical report, Workflow Management Coalition (1995)
8. van der Aalst, W., van Hee, K.: Workflow management : Models, methods, and systems. (2002)
9. Fischer, L.: Workflow handbook 2003. Future Strategies Inc., USA (2003)
10. Hillmann, D.: Using dublin core. Technical report, Dublin Core Metadata Initiative (2005)
11. Furrie, B.: Understanding marc - bibliographic machine readable cataloging. Technical report, Library of Congress - Network development and MARC standards office (2003)
12. Scansoft: Scansoft omnipage pro web page. (Web page) http://www.scansoft.com/omnipage.
13. RAG: Real academia galega web page. (Web page) http://www.realacademiagalega.org.
14. RAG: Real academia galega digital library. (Web page) http://www.realacademiagalega.org/Hemeroteca.

English Speech to Sanskrit Speech Translator

Abhradeep Guha Thakurta[1] Ashwath Kumar.K[2] and Hashir Khan[3]
guhathakurta.abhradeep@gmail.com[1] kashwathkumar@gmail.com[2]
hash9r@gmail.com[3]

National Institute of Technology Karnataka, Surathkal, India

Abstract. This paper presents a design level detail with an implementation insight of an English to Sanskrit Speech Translator that obtains a translation accuracy roughly 80% which is at par with similar systems developed using interlingua-based approach. The system uses both an Adaptive approach and a Non-Adaptive approach to perform the process of Language translation. The adaptive tools include Hidden Markov Model and Maximum Entropy framework whereas the Non-Adaptive techniques include Paninian grammar based parse tree mapping, techniques to restructure the words in the target sentence to conform to Indian Languages sentence structure, Longest Common Sequence based syllable splitter and heuristics based Text to Speech translator. Though the adaptive techniques that have been used are readily available but the total set of Non-Adaptive techniques and the method of integrating all the components is a unique design technique framed independently by us. The observed translating time of the system is approximately linear with respect to the input sentence. Furthermore, the system is able to translate sentences spoken in normal human speed with a rationally minimal time delay. For this reason this it can be used for online speech translation for all practical purposes. Keywords: Paninian, Longest Common Sequence, Parser,Maximum Entropy, Hidden Markov Model.

1 Introduction

Sanskrit has a very strong grammar structure and is the origin of many Indian languages. This makes it an excellent networking language which would facilitate the extension of the translator to other Indian languages. A translation scheme of this sort will have limitless applications in a land of such great diversity. Not only can this be extended to other Indian languages, a similar scheme may be used to create a translator from Indian languages to English through Sanskrit. The system under consideration is essentially grounded on four basic computational models. First, the HMM (Hidden Markov Model), Second, Maximum Entropy Model, Third, Paninian Grammar and Lastly on the varga based syllable joining and smoothening. The system is segregated into three parts, namely Speech Recognizer, Text to Text Translator (English to Sanskrit), Text to Speech Converter.

The three modules are finally integrated to run in parallel (with locks being implemented into them to take care of synchronization) in order to boost up

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 363–373

Received 17/06/07
Accepted 31/08/07
Final version 22/09/07

the performance. A detailed discussion on each of these sections will be dealt throughout the course of this paper.

The Online Speech recognizer uses Hidden Markov Model[1] as the Finite State Automata for recognition of the words. It essentially uses Tied State based Silence model[1] with a special short pause model also called the tee model. The signal conditioning and the signal analysis is done with the help of Triangular Filters MFCC (Mel Frequency Cepstrum Coefficient).

The English to Sanskrit Tex translator essentially banks on Maximum Entropy based English Parser[2] and Context Free Paninian grammar [3]. The parser has a parsing accuracy of 86%. The parser runs linear with respect to the sentence length. The context free grammar based Parse Tree Mapper for English to Sanskrit uses essentially the grammar foundation laid by Panini. We assumed the language to context free mainly because of the fact that English is proven to be 85% context free, so for all practical purposes we can assume a context free scenario. We have devised a unique technique to restructure the Sanskrit sentence to follow Indian Language form which essentially makes the language nice to comprehend. We have used the most prominent classes of sentences as background for designing. Though the class list used is not exhaustive, further additions can be made with ease due to the open architecture of the system.

Sanskrit is a phonographic language, so a phonological breakup of the words is a very feasible option for deriving the underlying syllables. We use our self designed algorithm for splitting up of words into their syllables. The algorithm is mainly based on Longest Sequence Match. This whole module is independently designed by us and it has accuracy which is comparable to systems like FESTIVAL and MBROLA. For all practical purposes we have used ITRANS and C9 tagset as the language representation format.

2 English Speech to English Text Conversion

The baseline system uses the HMM-LR method, which is an integration of Hidden Markov Models [1] and generalized LR parsing [1]. In this method, the LR parsing table is used to predict phoneme candidates in speech, then these phoneme candidates are verified using the HMM phoneme models. Thus, the LR parser can be regarded as a language source model for wordtphoneme prediction/generation. The following describes the basic mechanism of HMM-LR. First, the system picks up all phonemes predicted by the initial state of the LR parsing table and invokes the HMMs to verify the existence of these predicted phonemes. During this process, all possible partial parses are constructed in parallel. The HMM phoneme verifier receives a probability array which includes end point candidates and their probabilities, and updates it using an HMM probability calculation process. This probability array is attached to each partial parse. When the highest probability in the array is lower than a threshold level, the partial parse is pruned. The parsing process proceeds in this way, and stops if the parser detects an accept action in the LR parsing table. A very accurate,

efficient parsing method is achieved through the integrated processes of speech recognition and language analysis.

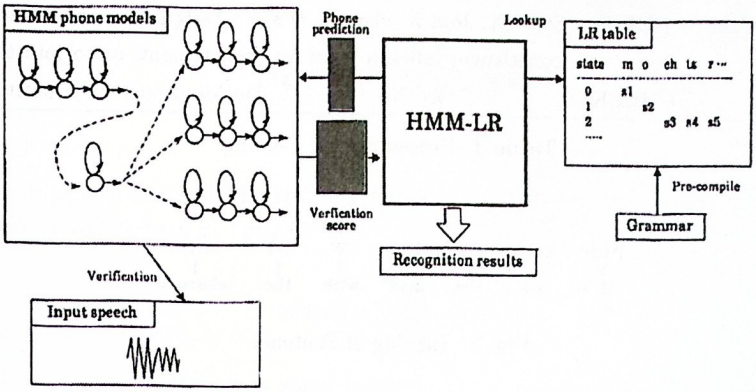


Fig. 1. HMM and LR based Automatic Speech Recognizer

3 English Text to Sanskrit Text Conversion

The English to Sanskrit Text Translator module is composed of three major sections. First, the Maximum Entropy based Parser, Second, the Parse tree Mapper between English and Sanskrit and Third, the positioning of the Sanskrit words to match Indian Language grammar form. The essence of the first two sections is inherent in the problem itself. But the third section is more of a nice to have feature than a must have feature. Sanskrit being a position independent language it is really not necessary to have a rigid positioning standard. But in most of the Indian scripts it follows a particular positioning norm which is equivalent to any Devanagari language (including Hindi). The inclusion of the positioning section is mostly to bring a smoothening effect in the language.

3.1 Maximum Entropy Parser

The parser uses four procedures, TAG, CHUNK, BUILD, and CHECK, that incrementally build parse trees with their actions. The procedures are applied in three left-to-right passes over the input sentence; the first pass applies TAG, the second pass applies CHUNK, and the third pass applies BUILD and CHECK. The passes, the procedures they apply, and the actions of the procedures are summarized in Table and described below.

First Pass The first pass takes an input sentence and uses TAG to assign each word a POS tag. The result of applying TAG to each word is shown in figure 2.

Pass	Procedure	Actions	Description
First Pass	TAG	A POS tag in tag set	Assign POS Tag to word
Second Pass	CHUNK	Start X, Join X, Other	Assign Chunk tag to POS tag and word
Third Pass	BUILD	Start X, Join X where X is a constituent label in label set	Assign current tree to start a new constituent, or to join the previous one
	CHECK	Yes, No	Decide if current constituent is complete

Table 1. Procedures for Parsing

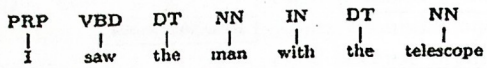


Fig. 2. Tagging of Sentence

Second Pass The second pass takes the output of the first pass and uses CHUNK to determine the "flat" phrase chunks of the sentence, where a phrase is "flat" if and only if it is a constituent whose children consist solely of POS tags.

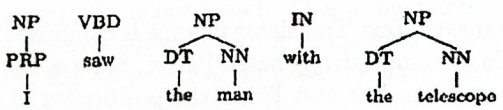


Fig. 3. After Chunking of the Sentence

Third Pass The third pass always alternates between the use of BUILD and CHECK, and completes any remaining constituent structure. BUILD decides whether a tree will start a new constituent or join the incomplete constituent immediately to its left. The result is shown in figure 4

3.2 Maximum Entropy Framework

The predicates for TAG, CHUNK, BUILD, and CHECK are used to scan the derivations of the trees in the corpus to form the training samples τ_{TAG}, τ_{CHUNK}, τ_{BUILD} and τ_{CHECK} respectively. Each training sample has the form $\tau = \{(a_l, b_1), (a_2, b_2), ..., a_N, b_N)\}$, where a_i is an action of the corresponding procedure and b_i is the list of contextual predicates that were true in the context

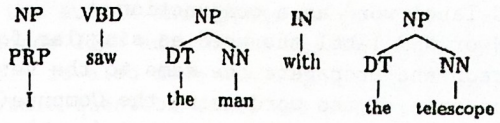


Fig. 4. After the third pass

in which al was decided.The training samples are respectively used to create the models τ_{TAG}, τ_{CHUNK}, τ_{BUILD} and τ_{CHECK} all of which have the form:

$$p(a,b) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(a,b)} \tag{1}$$

where a is some action, b is some context, is a normalizing factor. a_j are the model parameters, $0 < a_j < \infty$, and $f_j(a,b)\ \epsilon\{0,1\}$ are called features, $j = 1...k$. Features encode an action $a^{'}$ as well as some contextual predicate c_p that a tree-building procedure would find useful for predicting the action $a^{'}$. feature f_j: $f_j(a,b) = 1$ if $c_p(b) =$ true && $a = a^{'}$, 0 otherwise for use in the corresponding model. Each feature f_j corresponds to a parameter a_j, which can be viewed as a "weight" that reflects the importance of the feature. For each model, the corresponding conditional probability is defined as usual

$$p(a|b) = \frac{p(a,b)}{\sum_{a' \epsilon A} p(a^{'},b)} \tag{2}$$

Based on this we use BFS(Breadth first Search on the constituent tree structures derived so far to get the parse tree with the maximum probability

3.3 Translation of Parsed English sentence to Sanskrit

Tagging based on the type In this section the words are tagged based on their part of speech. We identify case, person, number and tense of the words in this function and tag the respective words for the further translation at a later stage. This tags are stored as additional information in each node of the tree as labels. This tagging is essentially on the basis of Sanskrit grammar. The tagging proceeds based on the tags assigned at the English Parsing stage.

Algorithm: Sanskrit Tagger (root)

```
//If the tag is DT label the current word and propagate the person to
the related verb.
//If tag is WP or WRB or any other question sentence. Label the word
as third person and propagate the tag to the related verb.
//If tag is JJ label word as adjective.
```

```
//If tag is CC label word as a conjunction.
//If tag is NN or NNS label the word as singular for NN and plural for
NNS, third person and propagate the same to the verb related to the
noun. Label the case of the word using the Compute().
//If tag is VBZ or VBP then label the word (verb) as present tense.
//If tag is VBD then label the word (verb) as past tense.
//If tag is MD then label the word (verb) as future tense.
//If tag is IN then label if the word is because tag as because
//If tag is VBN or VBG then retrieve the tense label from the first
related verb.
//If tag is PRP$ then depending on word label word as first, second
or third person. Do the same for singular and plural. Also label word
as shashti (case).
//If tag is PRP extract case using Compute(), check person using CheckPerson
```

Propagate the same to the related verb. This section decides the person of the noun and the pronoun. The return values are First, Second and Third coupled with Singular or Plural.

Algorithm: Check Person(Node)

```
// Check whether the node is of the type I, we or etc If yes then take
First. If You or yours then take Second. Otherwise take Third.
// Based of the number assign add Singular or Plural to the result.
```

This section tags the Determiner(DT). This is done after all the words are tagged.

Algorithm:

```
// Check if DT is subject.
// Check if it has a related noun. If so give determiner the label the
of noun.
// Else give the determinant its own label by making it First Case.
//If DT is object perform same steps as above. Case is found by using
the Compute( Node ) in case no noun is associated.
```

The Compute() function is used to determine the case of the word based on the verb using standard Paninian Grammar rules.

Final Translation This section essentially deals with converting each word and phrase from the English sentence to its corresponding Sanskrit form following the tagging and labeling that has been done in the previous sections.

Algorithm:

```
  //With the word we find the root word in Sanskrit.
//The root word is found from a MySQL database. This also stores data
about where the suffix is stored.
//Based on suffix table and various tense, person, case, and number
```

```
labels we find the suffix and append to get the target word.
//For special words like Adjectives and Conjunction we perform a direct
translation. This is also done where the word is independent of the
labels like certain question words and determinants.
```

3.4 Restructuring of the Sanskrit sentence

The basic methodology we would be adopting is that any Indian Language is of the form <Noun Phrase> <Object> <Verb Phrase>. So the fundamentally we would be trying to restructure the Sanskrit parse tree generated by the previous section to meet this sentence form. Moreover during our course of work we have identified three major classes of sentences of English Language which have to be separately treated to achieve the goal of restructuring. In the following section we would be dealing with each of these class of sentences and their associated sub classes and conversion Heuristic is detail.

Restructuring of Verb Phrase The basic difference between the sentences in English and Indian Languages is that in English the verb is always at the beginning of the verb phrase (VP) where as in Indian Languages the object is at the beginning of the verb phrase. This is also true for Preposition Phrases (PP). An example of such a conversion is given in Figure 5.

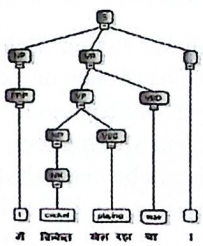


Fig. 5. Restructuring of Verb

Restructuring of WH-Phrase The class of sentences which encompass questions (essentially uses Who, What etc) are grouped together under the class of WH Phrases. In our dealing with this type of sentences we have found that the WH-Phrasecomes just before the principal verb in the Verb Phrase which it refers to. So in that case the sentences like Where were you? changes to you where were? which correctly maps on to is corresponding Hindi counterpart. We have used this technique for our restructuring.

Restructuring of No Sooner Than Sentences In computational linguistics this class of sentences is referred to as inverse sentences (SINV). This kind of sentences essentially has a to be verb (VBD) (i.e. did) and a principal verb phrase (VP). So the fundamental approach that is followed is removing all the nodes in the parse tree till the fundamental verb phrase and then placing no did just after the verb of the principal verb phrase. We have used this technique for our restructuring.

(N.B We do not claim that the classes of sentences that we have stated here is totally exhaustive. Its just a prototype and extension is always possible on similar lines.)

4 Sanskrit Text to Sanskrit Speech Conversion

Classical Sanskrit distinguishes 48 sounds. Some of these, are, however, allophones. The number of phonemes is smaller, at about 35. The sounds are traditionally listed in the order vowels (Ach), diphthongs (Hal), anusvara and visarga, stops (Spara) and nasals (starting in the back of the mouth and moving forward), and finally the liquids and fricatives. A database of the different syllables is generated. The database consists of three different recordings namely the

Vowels 15 in number
Half matras of consonants 25 in number
Semi Vowels 10 in number
Very commonly occurring syllables - 119 in number

A syllable frequency program was run for Mahabharatha and Sreebhyasa. All the syllables with a frequency of occurance of more than 1000 was recorded. It can be approximated that these are the commonly occurring syllables in Sanskrit. This number 1000 was just a heuristic number. If we increase the number of recordings, the quality of speech will be better but the space is a constraint. The labeling of the syllables is done using the ITRANS format.

4.1 Algorithm

The Procedure to be followed to convert the Sanskrit Text to Speech can be subdivided into the following steps. Since Sanskrit is a phoneme based language the conversion is easier. The input text is subdivided into smaller sentences which are again subdivided into smaller words. These words are pronounced using the pre recorded sounds of different syllables. The whole procedure can be epitomized into the following steps.

Tokenize the Sanskrit sentence to extract the words. A punctuation is used a delimiter to separate words.
Parse each word into syllables based on a syllable lookup database. The word is broken up based on the longest syllable match. The syllable database consists of the basic vowels, consonants and few commonly occurring syllables.

Join the sound files of the syllables to produce final sound of the word. Smoothening techniques are used to filter the output sound.
Join all the words with a silence (sil syllable) between them to get a continuous speech.

Tokenizing The basic idea in tokenizing is to separate a long sentence into smaller words and phrases. So every time punctuation is encountered a new token is obtained. Here Punctuations are used as delimiters to separate the sentences into words(tokens). So each of these tokens are passed on to the next stage for processing. So the Pronunciation is token by token.

Parsing During parsing each word is broken into different syllables present in the reference syllable database. The mapping of syllables is based on the largest syllable match. Initially depending on the syllables the program searches for the biggest common match between the token and the reference syllable database. During this process the program starts from the left end of the token and searches for the syllable with the largest match. Once it locates the largest match, it moves on to search the next largest syllable match in the token.

For example the word aasthaa becomes aa/s/th/aa where aa,s,th,aa are the different syllables present in the reference syllable database. `String function`

```
Parse(String s)
len←length(s);
parsed_string←NULL;
count←1;
while (count<=length)
{
//From the database find the longest syllable which is a prefix to the
string s[count  len] and store it in temp_str
parsed_string←parsed_String+temp_str+/;
count←count+length(temp_str);
}
//Remove the extra / from the end of parsed_string
return parsed_string
```

Pronunciation of each word In this module all the sound files of the corresponding syllables are added together to generate the sound file of the word. When combining the sound files of two syllables filtering techniques are used to reduce the transitions in the pronunciation of the word. To smoothen the transition between syllables, the last 100 samples of one sound is averaged with the first 100 samples of the next sound file and this replaces the 200 samples taken into consideration in the final output file. The value 100 is obtained by trial and error. The value acutually changes depending on the vargas of the two syllables.

For a better result, different values of sample merging can be used for different vargas and different combinations of the syllables.

The following plot (Figure 6) is obtained for the word haata. The plot is obtained using MATLAB. It can be observed the length of the first plot is around 12,300 samples, whereas the number of samples in the second case is reduced. Here the merging value used is 1000. In the second waveform it can be seen that theres a smoothened value near 8000 which is the transition from one syllable (haa in this case) to another syllable(tha in this case). An intermediate value is obtained near the transition which results in a smoothened sound.

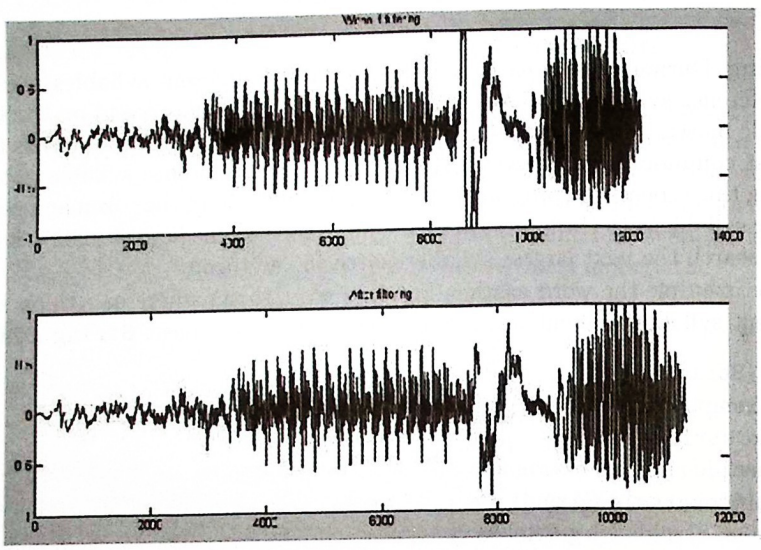


Fig. 6. Waveform for the word haata

Pronunciation of the whole sentence After each word is produced it is entered into the final sound file with a silence file in between to denote a punctuation. All the words produced are added to form a sentence by adding a pause between the words. Filtering techniques are used in this module also to reduce sharp transitions. By using the smoothening techniques the abrupt pauses are reduced and a smoothened and continuous sound can be obtained.

Filtering The filtering and smoothening of sound files while joining two sound files can be improved by using sophisticated filtering techniques. Here simple average of the two sound files has been used. Some better methods like Moving

Average Filter and Weighted filters can be used. By using complicated filtering techniques the computation time will increase affecting the overall time of execution. Even with the Simple Average Filter the overlapping sample count can be varied for different combinations of vargas. Here we have used a constant value of 100 samples for all the cases. By doing this the delivery of sound will have lesser transitions and the speech will be smoother compared to the original version.

5 Experiments

We have implemented the whole system in Microsoft Visual Studio 2003. We had tried on 35 different sentences with a word corpus of 100 words from the three different classes of sentences defined earlier. The results of the experimentation has given us a overall efficiency of 80% which is quite high keeping in mind the bottle neck situations of training the Speech Recognizer which drastically brings down the efficiency. The whole source code is available with the authors. Any one is welcome to use it.

6 Conclusion

As a part of final integration we have used a Shared Memory Model where where we run the three sections in parallel with locks implemented in them inorder to take care of empty or over flow of buffer. In a nutshell our effort has been to make a standing platform for further research in the same lines. Probably somewhere down the line a system can be made feasible which will do Real Time Emotion Based Speech Translation. If this is achieved the language barrier between Human races will be laid to rest forever. We hope for the best and encourage people to take up research work in this area to make world a better place to live in.

References

1. HMM Continious Speech Recognition using Stochastic Language Model.(Kenji KITA, Takeshi KAWABATA, Toshiyuki HANAZAWA ATR Interpreting Telephony Research Laboratories Seika-chou, Souraku-gun, Kyoto 619-02, JAPAN)
2. Linear Observed Time Statistical Parser Based on Maximum Entropy Models Adwait Ratnaparkhi, Dept. of Computer and Information Science University of Pennsylvania.
3. 3.Panini's Grammar and Computer Science Saroja Bhate and Subhash Kak,Annals of the Bhandarkar Oriental Research Institute, vol. 72,1993, pp. 79-94
4. ALTERNATIVES TO SYLLABLE-BASED ACCOUNTS OF CONSONANTAL PHONOTACTICS Donca Steriade, UCLA
5. OpenNLP resources at http://opennlp.sourceforge.net
6. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. Bo Pang, Kevin Knight, and Daniel Marcu. Proceedings of HLT/NAACL, 2003

Color Features Based Speaking Detection with Hidden Markov Model in Video Sequences

Peilin Jiang[1,2], Ran Li[2], Fuji Ren[1], Shingo Kuroiwa[1], and Nanning Zheng[2]

[1] The University of Tokushima, Tokushima, Japan,
jiang@is.tokushima-u.ac.jp,
[2] Xi'an Jiaotong University, Xi'an, 710049, China

Abstract. The Human Computer Interface Technology has faced challenges of understanding user's mind actively. In the first, the speak detection is a primary technique in applications of human computer interface (HCI) and other applications like surveillance system, video conference and multimedia data base management in computer vision and speech recognition. This paper describes a novel method to detect speaker with a probabilistic model of behavior of speaking. After human face recognition, the especial components under the nonlinear transformation in color space of lip represent the specific mouth region and then combine the groups of coherent motions . Next the simple movements in the mouth region are modeled by hidden Markov models. The experimental results demonstrate that the model representing speaking is efficiency and successful in applying to driver video surveillance system.

1 Introduction

The speak detection is one of the demanding tasks in the field of computer vision and speech recognition. It has played an important role in real-time and affective computing in HCI gradually before the recognition and synthesis procedures [2, 12]. For an instance, in the video conference, several participants discussed the problems together and the video camera needs to switch among the different speakers automatically. Thus, the recognition of speaker is a primary work before video data processing and transforming start. The same problem appears before affective computing in the human computer interface in which the speech and the expression is the important ground to determine the situation of emotion states transitions. The work has been firstly focused on the work of modeling the detailed mouth movement for lip-reading [13]. The common approach to speak detection is calculating movement energy function and then judging with an empirical threshold. Also, the information from audio is used too [6, 7]. However, the first problem, is that the threshold is empirical and dependent on individuality of different person and the second one is some video sequences which we need to process is asynchronous with audio sequences. To solve these problems, the previous works include template-based match, video-audio information fusion and supervised statistical learning models methods have been applied respectively. In this paper, an unsupervised human motion learning method, which do not

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 374–381

Received 16/06/07
Accepted 31/08/07
Final version 10/09/07

need mark training data ahead of time, is introduced to describe the behavior of speaking. The method estimates the motion as representation of a sequence of observations instead.

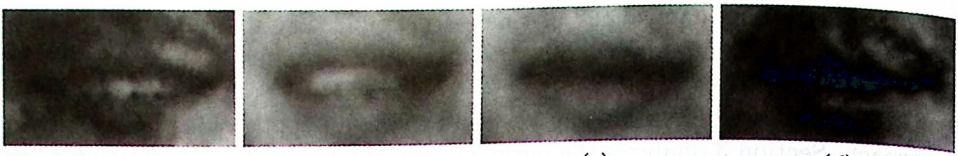
In Section 2 we discuss previous relative work and applications related to our approach. Section 3 analyzes the outline of the speck detection. In section 4, we describe the model of speaking and speak recognition, which includes prepare works before detection, color nonlinear map, and hidden Markov model learning algorithm. In section 5, we introduce the experiments of training the model and recognizing the speak behavior.

2 Related Works

Previously, the unsupervised learning approach for analysis of human motion was proposed by Tianshu Wang [3] to classify the uniform action from motion video sequences. Most earlier before that the research [14] had proved that the HMM model can be use to effectively recognize the human motions. On the other side, lip localization techniques also conclude a lot of results. Among them, an algorithm constructed mouth, eyes mapping based on light compensation and a nonlinear color transformation has been identified effective [4]. Additionally, several efforts are presented in speaker detection in many applications [1, 5, 8, 9].

3 Question Analysis

In the section 1, the problem is proposed that the description of speaking which we want to exploit is ambiguity because of noises, occlusion, random varieties from difference bodies. Hence, to a certain extent formerly mentioned methods are confused. The template-based method is more adaptive the localization problem than speak detection; the video-audio fusion method does not working well except the audio channel; the supervised statistical learning algorithms are effective but the data labeling is seriously heavy and hard to keep consistent. The traditional segment of mouth motion is in two phases which is showed in Fig.1. The open mouth and closed mouth respectively. Unfortunately, the borderline between these two phases is vague. Considering the speaking as the human complex dynamic such as walking and waving, we can decompose it with the probabilistic compositional framework[14]. Firstly, the work is divided into two steps: the mouth localization and speaking detection. In the localization step, the color feature in human face is extracted because of its unique property. Next, high-level complex speak motion is symbolized by Hidden Markov Model as successive phases of simple movements. Namely, the speaking is represented by a HMM model. And the HMM model is composed by the states which are observed as relatively simple dynamic models. These models are organized by temporal sequences of coherent motions that are described by low-level features. These underlying features are extracted from a sequence of input frames in which the color value of each pixel is random variable. Given a sequence of images $I_1, I_2 \ldots I_t$ includes HMM for motion, we can perform the recognition.



(a) (b) (c) (d)

Fig. 1. Examples of mouth states. The images in (a)and (b) are open mouth, (c) and (d) are close mouth.

$$P = P(HMM|I_1, I_2, \ldots, I_t) \tag{1}$$

The equation (1) presents the probability of HMMs in the image sequence. Several motions are corresponding to different HMMs.

4 Speak Model and Detection

Specifically, mouth speaking motion is represented here by a HMM model. Therefore, the detection task is converted into building the speak model and recognizing the HMM. The former is sum to calculate the optimal parameters of the hidden Markov model and then the second is estimation problem of optimal states sequences. In the section, we will introduce the processing of detection speaking with the HMM model. The first subsection describes the works before building speak model. In subsection 4.2, we describe a nonlinear transformation color feature being used as observations for coherent motion. Subsection 4.3 introduces the HMM learn model and detecting process.

4.1 Prepare Works

Before speak detection we must accurately localize the region of mouth in face area. In practical, the face detection algorithm is operated in the first step. In our work, we quote the cascade Adaboost based face detection algorithm [11] to execute the detection task. Successively the next phase is mouth localization. There are several fast and accurate algorithms, in which include color-based, texture-based, template-based and statistical learning. Compared with them we chose the luminance-chrominance space based mouth mapping for its successful stability under a wide arrange of conditions. After a light compensation, a novel nonlinear transformation is formulated to extract chrominance components which are sensitive on lip-tone region and skin-tone color. The details are introduced below.

4.2 Mouth Mapping based on Nonlinear Color Transformation

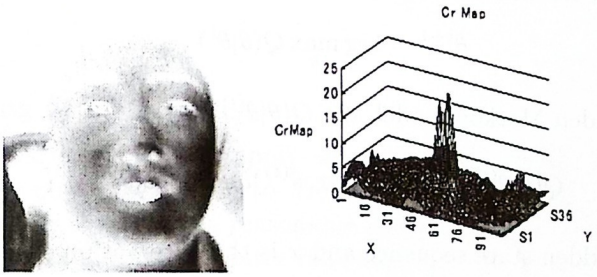
Modeling the human skin color requires firstly choosing an appropriate color space which cluster skin color together obviously. Among them, YC_bC_r space is

adopted because it is perceptually uniform and widely used in video compression standard and separable between components of luminance and chrominance [4]. The mouth region is special for its strong red color component and weak blue color component than other skin regions. These two factors correspond to the two components C_r and C_b respectively. The mouth mapping is then constructed nonlinearly:

$$Threshold_{mouth} = C_r^2 \cdot \left(C_r^2 - \eta \cdot \frac{C_r}{C_b} \right)^2 \tag{2}$$

$$\eta = 0.95 \cdot \frac{\frac{1}{n} \cdot \sum_{(x,y) \in \Im\Re} C_r(x,y)^2}{\frac{1}{n} \sum_{(x,y) \in \Im\Re} C_r(x,y)/C_b(x,y)} \tag{3}$$

The C_r/C_b feature and C_r^2 feature are found that low and high responses to the lip region in face respectively and both of them are normalized. n is the number of pixel within the face region. The h is estimated as a ratio of the average C_r^2 to average C_r/C_b The Fig.2 is the mouth map for the sample face.



(a) (b)

Fig. 2. Extracting mouth region by YCrCb map. The region with higher CrMap value in (b) is considered the mouth map.

4.3 Hidden Markov Model Learning

The hidden Markov model is statistically finite states machine with two sets of probabilities: a transition probability and an emission probability. For example, a HMM can be represented by a topology like in Fig. 3.

There are three basic problems of interest in HMM [15]. As the estimation problem, the parameters of hidden Markov model can be evaluated by EM algorithm [10]. In a probabilistic compositional framework, our goal is to classify the speak motion composed by relatively simple movements from the video sequence. We assume that the movements is sequential and then HMM can be modeled

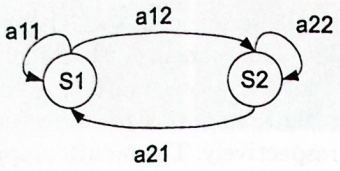


Fig. 3. Example of the two states HMM used for recognition

the motion. Each state corresponds to one simple phase. The EM algorithm provides a recursive estimation to converge on some limit of model parameters that achieve the local maximum.

In the E step, the Q function is calculated as following:

$$Q(\theta|\theta^{(t)}) = \sum_{h} P(h|v, \theta^{(t)}) logP(v, h|\theta)) \tag{4}$$

Where v is the visual part of the data and h is the hidden one.

In the M step, the maximum process is as below:

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta|\theta^{t}) \tag{5}$$

In the case of hidden Markov model, the $Q(\theta|\theta^{(t)})$ is:

$$Q(\theta|\theta^{(t)}) = \sum_{s} P(s|x, \theta^{(t)}) logP(x, s|\theta)) \tag{6}$$

Where s is the hidden state sequence and x is the observation sequence.

4.4 Model Recognition

In decoding problem, we aim to recover the sequence given an observation sequence. The Viterbi algorithm provides fast way of solving the decoding problem. The followings stare the Viterbi algorithm steps:

- Initialization: $\delta_1(i) = \pi_i b_i(x_1)$ and $\psi_1(i) = 0$ for all the states;
- Recursion: for $i = 2$ to T, j represents all states, $\delta_t = \max_{(1,N)}[\delta_t(i)]$, $\psi_t(j) = \arg\max_{(1,N)}[\delta_{t-1}(i)a_{ij}]$;
- Termination: $P = \max_{(1,N)}[\delta_T(i)]$ and $S = argmax_{(1,N)}[\delta_T(i)]$;
- Recover states:$S_t = \psi_{t+1}(S_{t+1})$, where $t = T - 1$ to 1;

However, although that for Viterbi only guarantees the maximum of over all state sequences and the resultant probability is only an approximation. The previous research shows that this is mostly sufficient. In here, we utilize this method as model recognition approach.

5 Experiments

In this paper, we build driver abnormal action surveillance system in order to detect speak motion to demonstrate the feasibility of model and recognition approach.

Our training data is 5 video sequences of 3 different individuals driving during when they make cell-phone call, which is considered to be dangerous, namely abnormal action during normal driving. An independent test data of a sequence also is ready for recognition experiment. Training data sequences has not been labeled and the chrominance component based motion features are used to localize the mouth firstly and then represent the motion features. The training and test data are all recorded in a steady background in either running car or stationary lab. Fig.4 shows some examples of one experiment video sequence.

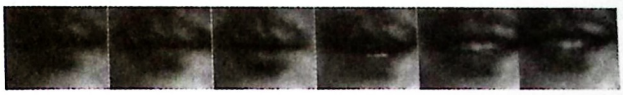


Fig. 4. Example frames of test data in training experiment.

5.1 Training Model

The training experiment is to apply with the test data that are not labeled which state the piecewise belongs to. We have tested difference numbers of states and conclude that 2 states is a reasonable opinion. We set the initial transition probability matrix and initial states randomly and train several times in order to make iterative learning procedure not converge into a local optimum.

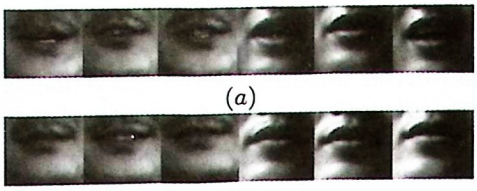
5.2 Recognizing the Speaking in Image Sequences

The final process was done to apply the learned HMM for recognition task on the test sequence. In term of the model and an unlabeled input sequence, we process the recursive decoding to obtain the optical states sequences. The sequence is split into pieces and the piecewise with high likelihood is the winner. In the result, the speak segment in test sequence is correctly recognized. The recognized segment is below in Fig.5:

In the next phase, we are going to performing more experiments on a large database.

6 Conclusion and Future Work

The human behavior like speaking, waving can be modeled by a probabilistic state transition model empirically. Like the other complex movements and gestures of human being such as walking and skating, the speaking can be clustered



(a)

(b)

Fig. 5. The example of detected speak video sequences. The upper row images (a)are segmented as speaking results, the bottom row images (b) are from no-speaking results

into certain sequent states corresponding to the observation sequences composed by low-level visual features and can be recognized separately. We conducted the training and testing experiments on video sequences of simulated driver action detection with the HMM and derived a satisfying result. Additionally, in a more general case, the observations in mouth region are a variant vector under the different poses of head and the improvement on the feature extraction and model parameter selection will be applied in the future work.

References

1. James M. Rehg, Kevin P. Murphy, Paul W. Fieguth: Vision-Based Speaker Detection Using Bayesian Networks. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99) - Volume 2 (1999)
2. Todor Ganchev, Nikos Fakotakis, George Kokkinakis: One-Speaker Detection-Limited Data: The WCL-1 System. NIST2003 Speaker Recognition Workshop, MD, USA (2003)
3. Tianshu Wang, Nanning Zheng, Yingqing Xu and Heung Yeung Shum: Unsupervised Clustering Analysis of Human Motion. Journal of Software (2003) 209-214
4. Rein-Lien Hsu, Mohamed Abdel-Mottaleb, Aril K.Jain: Face Detection in Color Images. IEEE Trans. Pattern Analysis and Machine Intelligence, vol.24, no.5, (2002) 696-706
5. Vladimir Pavlovic, Ashutosh Garg, James M. Rehg, and Thomas S.Huang: Multimodal Speaker Detection using Error Feedback Dynamic Bayesian Networks. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2000) 34 - 41
6. Ross Cutler and Larry Davis: Look who's talking: Speaker detection using video and audio correlation. IEEE International Conference on Multimedia and Expo (ICME), Manhattan, New York (2000)1589 - 1592
7. Nicolas Scheffer, Jean-Francois Bonastre: Speaker Detection using Acoustic Event Sequences. INTERSPEECH Lisboa, Portugal (2005)3065-3068
8. Mathieu Ben, Guillaume Gravier, Frederic Bimbot: A model space framework for efficient speaker detection. Proc. European Conf. on Speech Communication and Technology (2005) 3061C3064
9. Alexandre Preti, Nicolas Scheffer, Jean-Francois Bonastre: Discriminant Approaches for GMM Based speaker Detection Systems. MMUA, Toulouse, France (2006)

10. Rabiner Lawrence, Juang Biing-Hwang: Fundamentals of Speech Recognition. New York: Prentice Hall, (1993) 321-387
11. Paul Viola, Michael Jones: Rapid Object Detection Using a Boosted Cascade of Simple Features. IEEE Computer Vision and Pattern Recognition, Kauai Hawaii, USA. Volume 1.(2001) 511-518
12. Saitoh Takeshi, Kobayashi Harato and Konishi Ryosuke: Automatic speaker detection for videoconferencing. Technical report of IEICE. PRMU. Vol.106. No.72(20060518) (2006) 25-30
13. P Duchnowski, M Hunke, D Busching, U Meier: Toward movement-invariant automatic lip-reading and speech recognition. International Conf. Acoustics, Speech and Signal Processing, ICASSP-95 (1995)
14. Christoph Bregler: Learning and Recognizing Human Dynamics in Video Sequences. Proc. IEEE Conf. Computer Vision and Pattern Recognition (1997)568 - 574
15. Rabiner, L.R: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE. Volume 77, Issue 2, (1989) 257 - 286

Intelligent Tutoring Systems

Authoring Fuzzy-based Tutoring Systems for M-Learning

Ramón Zatarain, M. L. Barrón-Estrada, Guillermo A. Sandoval-Sánchez and L. Vega-Zazueta, Carlos A. Reyes-García *

Instituto Tecnologico de Culiacan, Juan de Dios Batiz s/n Col. Guadalupe, C.P. 88220
Culiacán, México, tel. +52 667 7131796
rzatarain@itculiacan.edu.mx
* Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, 72840, México
kargaxxi@inaoep.mx

Abstract. This paper is about an authoring tool that is being implemented to produce intelligent tutoring systems or any kind of m-learning application for different mobile devices like Cell Phones, Smart Phones and PDAs. The system is divided into three main components: The Editor of courses, which is an IDE (Integrated Development Environment) where the author create, open and save the user application; An export module to produce applications for different kinds of mobile devices and for different kinds of LMS (Learning Management Systems); and the intelligent module which adds adaptation features to the application produced by the user. Right now, the system has been tested by producing different application for Cell Phones and PDAs. We show some example of those applications.

1 Introduction

Whereas the presence of e-Learning is nowadays considered something common, it is not the same when we talk about the use of m-Learning. This is due to a couple of factors: first; we have that this technology was barely used, and second its use in the area of the education is still more recent. M-Learning is any service or facility that provides the student general electronic information and educative content that helps the process of knowledge acquisition without concerning the place or the moment [3].

E-Learning (electronic learning) has been accepted during the last two decades by several factors; one of them is the existence of students that by lack of time can not attend regular classes at an educational center; another one is that in certain regions, for some students, it is really hard to reach geographically speaking. During the time of life of e-Learning one of the bigger supports it has received has been from the Internet. The Web is a very rich atmosphere for educative investigation: the Web can provide several types of learning materials and given its powerful systems of data collection these can be added easily to web-based learning materials [4].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 385–394

Received 19/06/07
Accepted 31/08/07
Final version 20/09/07

On the other hand, technology has changed much in the last years, and, along with this advance, the costs of the devices have diminished. Computers have become mobile, and with it a new paradigm arrives to us: "anytime, anywhere computing" [1]. This becomes patent in different areas like commerce, entertainment, communications and industry of services, operations of businesses, learning and training [2]. In the field of learning, the benefits of this new mobility are expected to means a more efficient instruction and improvement in the results of the education process [1].

In the field of computer-based learning there exists the ITS (Intelligent Tutoring Systems). The ITS systems are software programs behaving in many aspects in a similar way to human tutors [9] and they try to imitate their activities. They are used in diverse areas, such as mathematics, physics and education of languages [10]. The popularity of ITS is growing day with day due to diverse factors like increasing the performance of the students, expanding the cognitive development, and reducing the time that the student requires to acquire new knowledge [11].

The task of creating courses for mobile devices is not a simple work, and requires technical knowledge of the computer science field (design, programming, etc.). That is why even when there are specialists in topics of different fields of knowledge willing to create mobile courses this is not possible given their lack of experience in the task of programming of this type of devices.

In order to deal with this new challenge of producing adaptive learning applications, we implemented an author tool where a non-programmer instructor can more easily produce Intelligent Tutoring System for mobile devices.

The arrangement of the paper is as follows: In Section 2, we describe the software architecture of the tool. In Section 3, we discuss the implementation of the software. Tests and results are presented in Section 4. Comparison to related work is given in section 5 and conclusions and future work are discussed in Section 6.

2 Multitutor Architecture

The application is divided in two parts; the first one is a desktop program, in which tasks such as creation, edition and publishing of the courses are made, while the other is an interpreter able to visualize the course material on the target mobile (see figure 1).



Figure 1. Desktop application and interpreter

Therefore we have a version of the desktop application that always makes the same tasks, and tends to need a smaller amount of changes by its static nature, and a mobile application, able to adapt to the capacities of the mobiles and type of learning.

In the architecture of the desktop application a model of layers is used, which has three levels; being the first of them the *user layer* where the graphical interfaces

appear, and where the user has a direct interaction with the author tool; the second level called *business layer* is the one that contains the logic of the application which makes all the tasks that the user asks to the system through the components that are part of the previous layer; whereas in the lowest level is placed the *data layer*, in which the components that carry out the task of saving information in storage media are found.

This architecture benefits the system design since it allows us to have a better organized application, with cleaner code, which tends to be free of errors and easy to debug since each of the main components (user interface , system logic and data persistence) are separated in modules.

On figure 2 we can observe the layer architecture of the desktop application. We show only a few components that are part of each level.

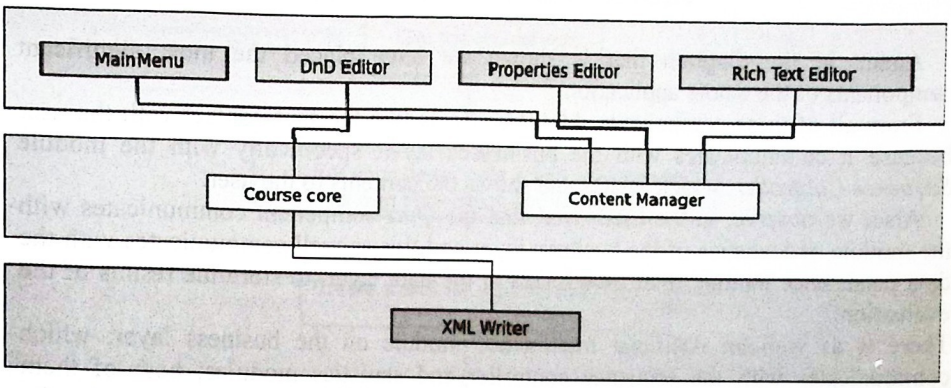


Figure 2. Desktop application architecture

We can see component *MainMenu*, which is used as the starting point in the application. From that menu, the application calls the course editor.

We also have the *DnDEditor*, which shows the course structure to the user, allowing him/her to edit the course using drag and drop actions.

On the business layer we have component *course core* used to keep communication with the *DnDEditor* component at the user level, and from where changes made by the course author are reflected. In other words, this module contains the course information.

XMLWriter is a component of the data layer which allows accomplishing the task of saving the course in storage media. Also, this module is the one that cooperates on the needed tasks to publish the courses.

As we can observe on figure 2, each of the layers can only communicate with the contiguous layer, respecting in this way the layer model, and favoring encapsulation.

With regard to the interpreter for the mobile device we can see the architecture in figure 3. It has been chosen, like in the desktop application architecture, a model of layers. This architecture has been chosen for being fast to implement, easy to debug and to favor its maintainability, being this a critical point in the application.

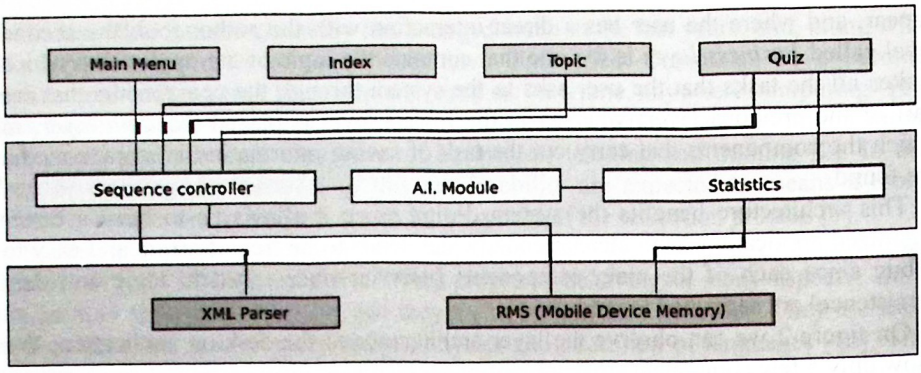


Figure 3. Mobile interpreter architecture

Again, in the diagram that is shown we have placed the most significant components of the whole application.

From all of these components, *MainMenu* acts like the first contact with the user because it communicates with the businesses layer, specifically with the module *Sequence Controller*, which selects and shows the contents to the user.

Also, we observe, in the user layer, that the *Quiz* component communicates with the module of Statistics of the business layer and this as well communicates with the data persistence module (that uses RMS) in the data layer, to store the results of the evaluation.

There is as well an Artificial Intelligence module on the business layer, which communicates with the sequence controller and statistics modules; both of them provide the intelligent module with information about the user interaction with the course; with that information this module performs several tasks (using fuzzy logic techniques) to determine which information is shown and how it is going to be shown to the user according to the type of learning.

3 Implementation

Multitutor Móvil is a multiplatform independent implementation which uses Java and XML technologies. Java has been chosen as the language for the desktop application to take advantage of the multiplatform support, which helps the application to work on a wide range of computers without requiring neither complete nor partial reprogramming.

In order to implement the persistence of data, XML has been used, because of its flexibility for generating future extensions to the project; besides XML provides standard ways to communicate with other applications.

We need to mention that when a user decides to initiate the authoring of a new course; a user interface is shown from which the structure of the course must be created, as well as the existing relations between each one of the objects that compose it. This task is made through an action of drag & drop.

When the user finishes the edition of the course he/she continues with the exporting process. This task at user level is carried out with only selecting the export option at the project menu.

For implementing those operations, the first step is finding the root of the course (which internally is represented like a tree). Once it has been located the program begins to traverse the tree and to extract the contents and properties of each of the nodes (which are elements of the course). Each of the elements is written on one or several XML files which reflect the information and the attributes of them.

When the process of writing all of the XML files has finished, it proceeds to create a package, by compressing the information into a *JAR* file and adding the mobile device interpreter to it. Figure 4 illustrates the process made by the exporting course algorithm.

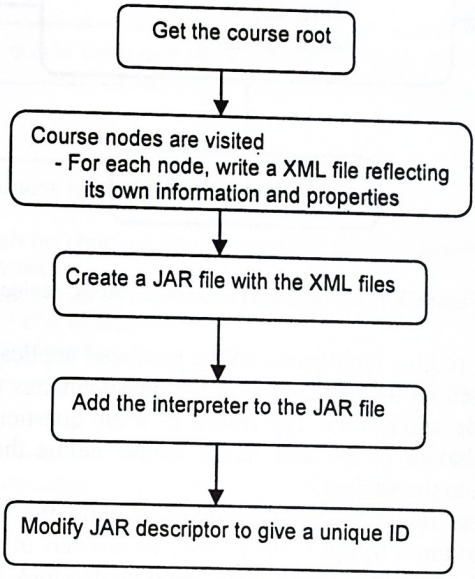


Figure 4. Exporting course algorithm

After exporting a course we need to copy it to the mobile device memory for its installation. The process of copying is made in different ways depending on the mobile device and the operating system running in the desktop computer. Installing the software provided by the manufacturer of the mobile device is usually enough to accomplish the communication with the computer.

Once the program is in the mobile device, the next step is to install it. When the process is over it will be possible to locate it in the list of applications that the device recognizes as installed in its memory.

For executing the course the interpreter takes control of this process by reading and showing the contents of the XML files (see figure 5).

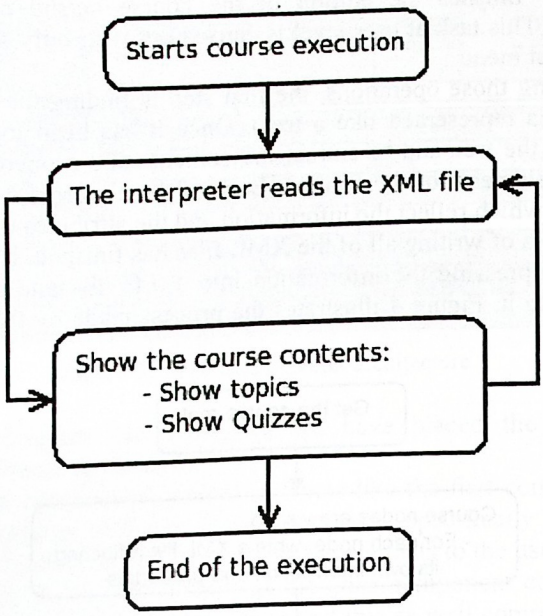


Figure 5. Interpreter execution on a mobile device

There is a module to give intelligence to the produced applications. The operation of that module is based on the application of the questionnaires that previously were defined in the desktop application. The results of these questionnaires, added to an evaluation of the behavior of the user in the course define the way in which the information is shown to the student.

We defines several linguistic variables like the time the user spends in each subject, the number of times he opens the course, the answers in questionnaires (if the correct answer was the first or second choice, the time that took the user to answer it, etc.). Fuzzy values are given to each of the evaluated points or linguistic variables. We assign weighted values to each variable involved on the choice of the course path to be followed. In other words, our student model is like a two dimensional array of linguistic variables, corresponding to different combinations of intelligences (logical/mathematical, linguistic/verbal, visual/spatial, etc.) and learning goals, where we stores the weighting factor that indicates the students preferred learning characteristics. The information stored in the student model includes the navigation history, the time spent on each learning unit, the answers given to questions and the different sections covered in the learning units.

For example, a student will receive a first quiz with 18 questions. Each question has its weighted value for each type of intelligence where the weight is assigned depending on the answer. For each question two possible answers exist that are designed to be opposite among them. In case that the user does not respond a question we will not compute this question.

Our formula for computing the type of intelligence to a given user is:

If (user shows a linguistic answer in question$_m$) then
LinguisticVar = LinguisticVar + (((LinguisticVar * value of question$_m$))*100)/ # (of linguistic answer number)

Later, the student is evaluated again but now according to different variables like spent time in a subject or unit, path the student choose, number of times he spent in a given subject, and other quizzes he/she takes at the end of a subject or unit. An important point here is that the author (an instructor) decides the elements of evaluation, the normal time in a topic, and other variables that are used for building the most preferred type of learning of any student.

The course of action in which the course adapts to the user intelligence depending of different attributes obtained during the learning process is illustrated in figure 6.

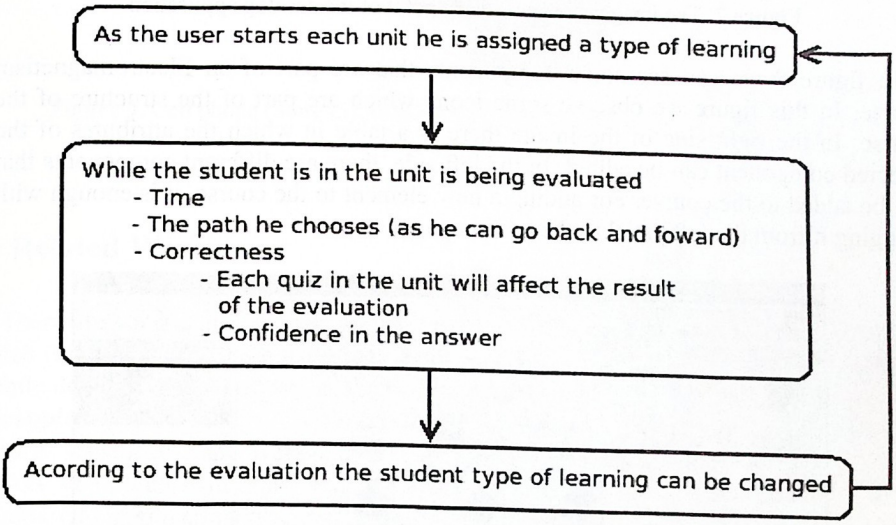


Figure 6. Learning type selection algorithm

4 Test and Results

In this section, we present some figures that illustrate the main interfaces of the Multitutor author tool. Beside we give an example of an application created with the tool. As we can see, Figure 7 shows the interface of the main menu. We can see several options, such as the creation of new courses, or opening some of the existing ones for its edition or publication.

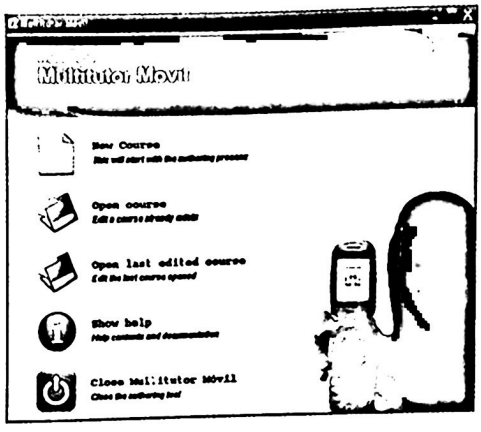


Figure 7. The image shows the entry point of the desktop application.

In figure 8 we can see some components that are part of an Electromagnetism course. In this figure we observe some icons which are part of the structure of the course. In the right side of the image there is a table in which the attributes of the selected component can be edited. In the left side, there are different components that can be added to the course. For adding a new element to the course, it is enough with dragging it from the left panel to the edition panel.

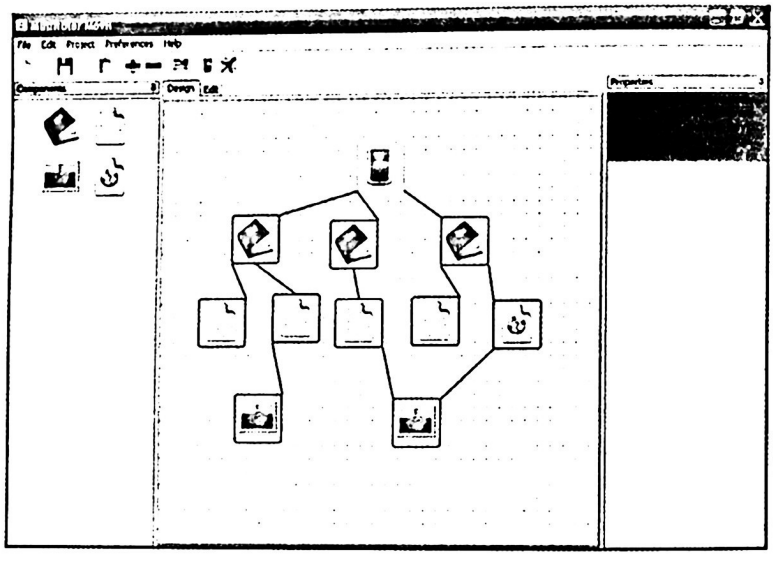


Figure 8. The structure of an electromagnetism course

A course on electromagnetism was created and distributed with students of the Instituto Tecnológico de Culiacán. Students used the course as a source of information for reviewing the topics of the course. The course was tested and successfully ran in different devices, such as PDA's, PocketsPC's and cell phones owned by the students and instructors of the institution.

Figure 9 show part of the course in different mobile devices.



Figure 9. Cell phone (Sony Ericsson k750) running an electromagnetism course

5 Related Work

There are several author tools used to create mobile applications. *MyLearning* [12] which is developed by *m-learning.org* is one of them. *MyLearning* is used for creating mobile device courses. However, the variety of mobile devices which are the target of this application are PocketPC's, which are a type of mobile devices that have technical features greater than the average, and represent a much smaller p of penetration in the market compared with the cellular. On the other side, *Hot Lava* [13], another commercial application oriented to m-learning, has as main disadvantage that the market of the produced applications does not include cell phones. In addition, both programs do not have the capacities of adaptation to the form of learning of the user.

6 Conclusions and Future Work

Multitutor is an author tool that allows an instructor to produce intelligent applications for mobile devices. The author can produce mobile an application from a desktop computer. The application can include different components like text, images, and questionnaires as well as learning features according to the type of intelligence of the student. One feature of the tool is simplicity, because it allows a fast creation of mobile applications. The software has been implemented in Java version 1.6 and XML. So, we can use Multitutor in any device where a J2VM is installed.

Right now we still are working and testing the intelligent tutoring module to each of the produced courses.

References

1. Luis de Marcos, José Ramón Hilera, José Antonio Gutiérrez, Carmen Pagés, José Javier. Martínez. Implementing Learning Objects Repositories for Mobile Devices. Department of Computer Science, University of Alcalá. Alcalá de Henares (Madrid). Spain.
2. Eilif Trondsen. Global Perspective on mLearning. SRI Business Intelligence. San Francisco. Febrero del 2005.
3. Lehner, F. and Nosekabel, H. The Role of Mobile Devices in E-learning - First Experience with a E-learning Environment. IEEE International Workshop on Wireless and Mobile Technologies in Education. Los Alamitos, USA: IEEE Computer Society. 2002
4. Harri Ketamo, mLearning for kindergarten's mathematics teaching. Tampere University of Technology, information technology, Pori. IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02). 2002
5. Connolly Thomas and Stansfield Mark. Using Games-Based eLearning Technologies in Overcoming Difficulties in Teaching Information Systems. University of Paisley, Paisley, Scotland, UK. Journal of Information Technology Education. Volume 5, 2006
6. Quinn, C. M-Learning. Mobile, Wireless, in-your-pocket Learning, http://www.linezine.com/2.1/features/cqmmwiyp.htm Marzo del 2005.
7. Houser, C., Thornton, P., Kluge, D., Mobile Learning: Cell Phones and PDAs for Education. Internartional Conference on Computers in Education ICCE'02. London, UK. 2002
8. Shepherd, C. M is for Maybe. http://www.fastrak-consulting.co.uk/tactix/features/mLearning.htm Marzo del 2005.
9. Edward R. Sykes, Franya Franek. Web-Based Architecture of an Intelligent Tutoring System for Remote Students Learning to Program Java.
10. Woolf, B., P., Beck, J., Eliot, C., & Stern, M. Growth and maturity of intelligent tutoring systems: A status report. Smart machines in education. Cambridge, MA: MIT Press.
11. Graesser A. C., Person, N. K., & Harter, D. (2001). Teaching tactics and dialog in autotutor. International Journal of Artificial Intelligence in Education, 12, 12-23.
12. Attewell, J (2005). Mobile technologies and learning: A technology update and mlearning project summary. London: Learning and Skills Development Agency. Available online at www.m-learning.org/reports.shtml. June 15, 2007.
13. McConatha, D., Praul, M. Mobile Learning in the Classroom: An Empirical Assessment of a New Tool for Students and Teachers, http://www.hotlavasoftware.com. June 15, 2007.

An Approach to Learning Objects Classification using an Auto-Associative Neural Network

Clara Pérez-Molina and Miguel Rodríguez-Artacho

UNED ETSI Industriales / UNED ETSI Informática
Juan del Rosal, 16 28040 Madrid (Spain)
clarapm@ieec.uned.es , miguel@lsi.uned.es

Abstract. In this paper we present an unsupervised neural network-based approach to multiword extraction and semantic classification of learning material with potential application to automatic metadata generation and ontological classification of learning objects. The described experience shows the behavior of a neural network-based system to extract semantic units in a given learning material. Semantic units contribute to a better indexing and therefore characterization of the dataset making easy the access to the knowledge contained in the course. Extraction of these semantic units identifies the content from an instructional perspective and can then be applied to instructional classification according to a given ontology.

1 Introduction

Meaningful words found together in a variety of contexts can express very accurately the semantic of a given text. These items, also known as multiword expressions or complex concepts, consist of two or more words that correspond to some conventional way of referring topics or concepts [1]. When we search information about a knowledge item, the subject is usually described by a set of words; this is what we call semantic unit. When applying semantic units' extraction to the educational content arena, we find that the help of existing ontologies and educational classifications can allow a framework to automatically scaffold chunks of content according to these conceptualizations.

In this context, classification of learning material is generally done providing a precise set of words that accurately describe a given resource. As we know, current authoring process of learning content is difficult and costly as it combines search and retrieval of the appropriate Learning Objects (LO) to compose larger units of content [9]. Current standards as IMS or SCORM label with a rich metadata profile the new resource which is manually elaborated and this result in a trade-off between low granularity resources, very reusable, but costly and major pieces not as reusable due to high context dependant information.

Thus approaches to automatic semantic labelling of content are becoming an important objective in order to achieve automatic metadata generation as current model of manually labelled material has been frequently reported to be difficult to maintain [7,8]. Some approaches have been developed by means of clustering of

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 395–405

Received 17/06/07
Accepted 31/08/07
Final version 30/09/07

learning material using formal concept analysis [10], which eventually can capture the knowledge representation of the learning content as a lattice, similar to a real taxonomy.

In this paper we propose to apply a parallel approach using an associative neural network self trained with non supervised learning to the classification of a given educational material. Our aim is to explore non supervised training of our neural network with the intention to obtain a set of semantic units that could describe any educational material and classify it according to a given instructional classification, as for instance the one developed in [9]. The rest of the paper is structured as follows: section 2 explains our neural network design and the semantic units' extraction. Section 3 applies the model to a given course, extracting the neural network results. Finally section 4 shows the conclusions.

2 Classification System Design

At first glance, the simplest method for finding out semantic units is counting, therefore, in many cases researchers rely on frequency methods. However, neither just selecting the most frequent bigrams (sequences of two adjacent words), nor considering sets of part of speech tag patterns (Adjective-Noun, Noun-Noun, Adjective-Adjective-Noun,...) are good solutions, due to the noisy words that sometimes appear between connected words [2]. On the other hand, syntactic methods need a previous tagged phase, which sometimes is not possible to carry out and, moreover, it may introduce errors to the rest of the process [3].

The nature of management information problem in text documents has provided an important area of application for some kinds of neural networks [4]. Unsupervised neural networks as Kohonen networks [5], Hopfield networks or hybrid systems [6] have been implemented.

Our intention is that the developed system generates behavior pattern matrixes from term sequences in documents that are used to train an auto-associative neural network. After the training stage, the network owns information that involves every atomic concept appeared in the text. The analysis of neuron connections provides a method for detecting semantic units inspecting the heaviest weights in the network, since they are consequence of most repeated behavior patterns. In this way, the model allows to identify sequences of words that represent semantic units in the dataset processed.

2.1 Generation of Patterns

Each neuron in the network represents one atomic concept, that is, a concept that can be represented by just one content word. In order to carry out this correspondence between neurons and atomic concepts, a previous analysis stage is required. Such analysis stage has two objectives. The first one is to identify which words can represent atomic concepts. The second one is to ensure that whenever a particular concept appears it is always represented by the same neuron. In order to do that, a

stemming process is applied to the content words found in the text. (See Figure 1, where only first 100 of 487 files processed is shown)

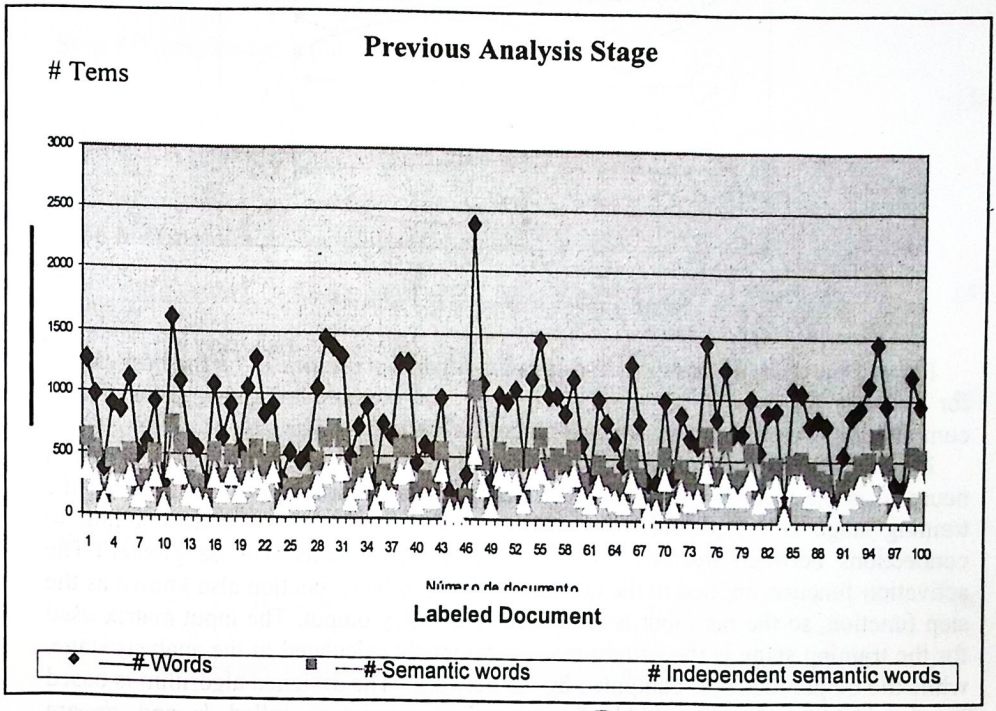


Fig. 1. Previous Analysis Stage: Independent semantic words identification.

The pattern generation phase starts once atomic concepts are identified in the text. In, order to generate the behavior pattern matrix associated to the text, word environments are considered through a scrolling window that is moving over the text. The window size is related to the maximum number of atomic concepts that constitute semantic units detected by the system; however this parameter can be easily modified.

After concluding this stage, we obtain a behavior matrix M that represents the entire text, whose dimensions are defined by SxQ. S is the number of atomic concepts found in the text and Q is the number of generated patterns. The set of patterns are grouped according to their generation order to shape the column vectors of M, taking into account that each pattern is originated by one scrolling window position.

2.2 The Neural Network Algorithm

The neural network architecture selected for the system is a fully connected neural net. Hence, each neuron is connected to the rest of them, including itself.

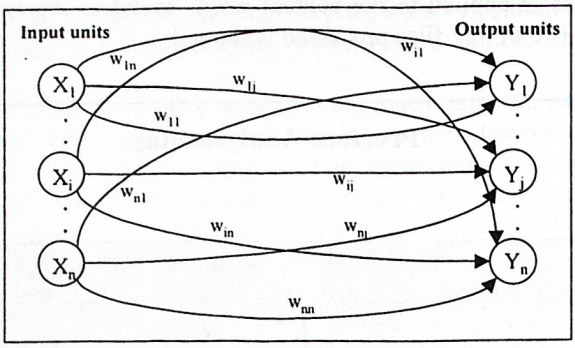


Fig. 2 Autoassociative neural network.

For our purpose, an unsupervised learning algorithm seemed to be the best choice for training the neural network, since it is not possible to know, prior to the computation, which is the appropriate output for the net.

Initially the weight matrix, that reveals the strength of connections between neurons, is the identity matrix, so each neuron is just associated to itself. When the training stage is completed, the elements of the weight matrix corresponding to connections between neurons that make up semantic units will be greater. The activation function applied to the network is the hard limit function also known as the step function, so the net input is converted to a binary output. The input matrix used for the training stage is the pattern matrix previously calculated in the analysis stage, while the output matrix is computed by the network. The designed algorithm is based on the Hebbian learning [11]. Two learning parameters called *lr* and *dr* are introduced. The first one represent the learning rate, the second one the decay rate. These parameters can be adjusted to achieve the best configuration according to the dataset. The developed algorithm for an autoassociate neural network of S neurons can be described by the following steps:

Step 0: Firstly, it is necessary to calculate the dimension of the network. Considering M as the representative matrix of the text, there are as many neurons as rows, S.

Step 1: The weight matrix of the net $W=\{w_{ij}\}$ is initially the identity matrix.

$$W = W_0 = I \qquad \begin{matrix} w_{ij} = 0 & \text{if } i \neq j \\ w_{ij} = 0 & \text{if } i = j \end{matrix} \tag{1}$$

Step 2: A bias term, whose value is controlled through a parameter α, is introduced. The bias term B is expressed by a vector with as many components as neurons constitute the network. Initially,

$$B = B_0 = -\alpha \, \mathrm{u} \qquad b_i = -\alpha \tag{2}$$

Step 3: For T times, do steps 4-7: ($\tau = 1,...,T$)

Step 4: For each of the Q training vector P(q) do steps 5-7: ($q = 1,....,Q$)

Step 5: Compute net input,

$$Y_in(q) = W * P(q) + B \qquad y_in_i(q) = \sum_{j=1}^{S} w_{ij} * p_j(q) + b_i \tag{3}$$

Step 6: Determine activation (output signal),

$$A(q) = f(Y_in(q)) = hard\lim(Y_in(q)) = hard\lim(W * P(q) + B) \tag{4}$$

For each unit, the activation is

$$a_i(q) = hard\lim\left(\sum_{j=1}^{S} w_{ij} * p_j + b_i\right) \tag{5}$$

Step 7: Update weights,

$$W^{\tau} = W^{\tau-1} + dW^{\tau} \qquad w_{ij}^{\tau} = w_{ij}^{\tau-1} + dw_{ij}^{\tau} \tag{6}$$

The weight matrix W of the net changes by an amount dW given by,

$$dW^{\tau} = lr * A(q) * P^{T}(q) - dr * W^{\tau-1} \qquad dw_{ij}^{\tau} = lr * a_i(q) * p_j(q) - dr * w_{ij}^{\tau-1} \tag{7}$$

Training stage is finished when the number of iterations has reached a determined value T, this parameter needs to be adjusted for each dataset. Information about atomic concept connections is then contained in the weight matrix of the network, so that it is necessary to develop an analysis mechanism to find them out.

2.3 Semantic Units' Extraction

The analysis method applies a filter process to the weight matrix, so that every element not exceeding a threshold μ is rejected. Elements belonging to the main diagonal are also removed in the filter process, since they do not involve information (according to our exposition, elements of the main diagonal represent the relationship between one atomic concept and itself) (Figures 3 and 4).

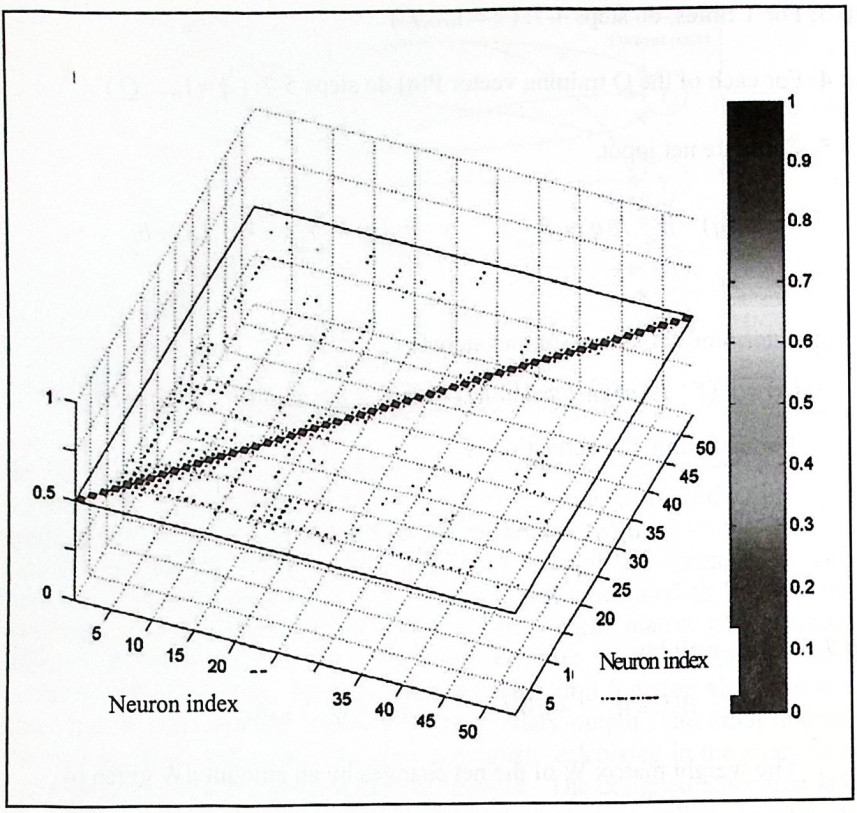


Fig. 3. The figure shows the visualization of the weight matrix. In weight matrix representation, x and y axes represent neuron indexes and z axe the corresponding weight for each connection. This example is extracted of a network of 53 neurons.

When the filter process to the weight matrix is concluded, the extraction of semantic units' stage begins

The extraction stage is carried out in different phases; each one is in charge of the extraction of semantic units of a particular number of atomic words. The developed system is able to extract semantic units made of two, three, four and five atomic concepts. First of all, the system search for semantic units made of two terms, and after the number of terms is increased one by one. The search algorithm takes into account that the weight matrix is not symmetric. Due to learning algorithm characteristics used for training the neural network, the weight on connection from neuron i to neuron j is different from the weight on connection from j to i. Therefore, the search algorithm imposes the criterion of reciprocity. It establishes that if neuron i and j make up a couple it is necessary that both weights w_{ij} and w_{ji} exceed the thresholdμ. When couples are extracted, the search algorithm concerns with complex concepts of three terms. The algorithm assumes that every concept belonging to a set of three elements makes up couple with the rest of them. Finally, the extraction of semantic units of four and five terms is carried out in the same way.

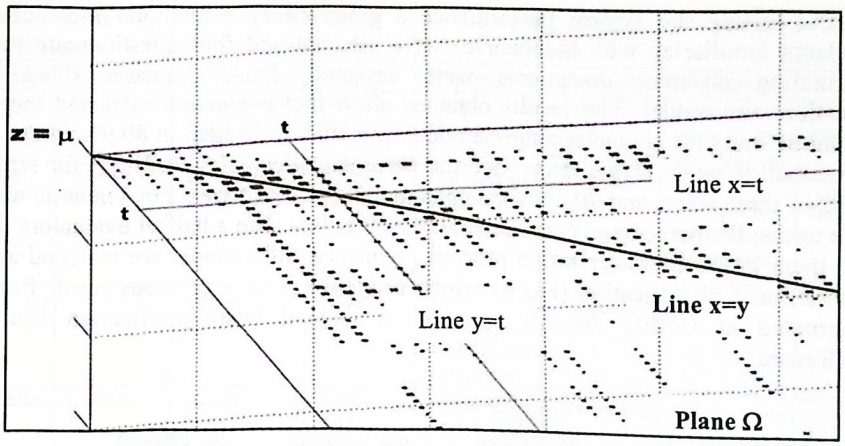


Fig. 4. The figure shows the visualization of the filter process as the projection of a three dimensional space representation of the weight matrix over the plane z=μ; x and y axes represent neuron indexes.

3 An Approach to Classify Learning Material: Experimental Results

The data used in our experiments consists of a collection of 487 documents concerning to different Java aspects of a generic Java course[1] arranged as a freely available web site.

Taking into account the formatting stage applied to the files, the semantic units extracted are made up of terms that do not appear in the text in the same form (or at least the majority of them). The identification process of the semantic units associated to each document of the dataset, requires an additional process by which the system applies an ordering method and assigns a linguistic expression to each group.

	# words	# Filtered words	# indep. terms	# dependent terms	# patters	Document Representative matrix dimension
Arithmetic mean	153	69	48	36	79	48 x 79
Median	136	62	46	29	71	46 x 71
Standard Deviation	67	34	16	23	36	16 x 36
Variance	4535	1134	282	423	1312	282 x 1312

Fig. 5. Semantic units' results classified by words, terms and patterns

[1] In this experiment our source is "Introducción a Java" course. We thank DISPEIL project, UNED and Universidad Carlos III de Madrid.

For testing the system performance a group of 12 evaluators (most of them students familiarize with the course) were selected and four questionnaire models containing different documents were designed. Each evaluator filled in a questionnaire model. The results obtained show that evaluators extracted the same semantic units for characterizing the documents that the system in 80.9% of the cases for semantic units of two terms, and the percentage decreases to 71.2% for semantic units of three terms and 70.4% for semantic units of four terms. For semantic units of five terms, the percentage exceeds 60%, however less than a half of evaluators do not use them in the characterization process. Semantic units model are matched against instructional classification (Fig 6) which corresponds to a previous result from [9] elaborated at UNED. In this approach a manual post-classification has been performed.

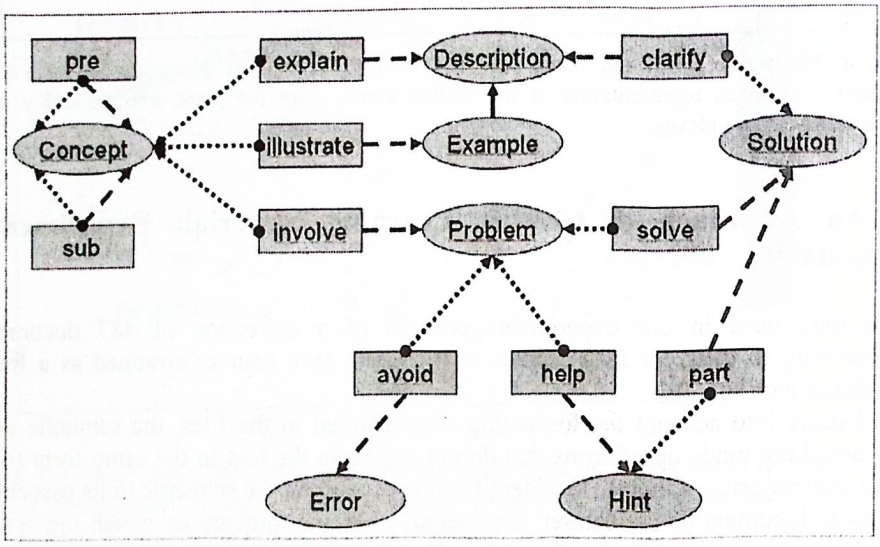


Fig. 6. The existing instructional conceptualization

Multi-words consistent of course related terms are searched in a Java vocabulary and then manually classified in the ontology. The main interest has not been so far to perform at this stage a full automatic process of classification but to explore how accurately semantic units' extraction can be mapped against an ad-hoc classification [14].

For instance, let us consider an example obtained by processing through the semantic network a given file. Processing results are shown in Table 1 where semantic units of size 1 to 5 are shown. Multi-words consistent of java related topics (as reserved java words like `for`, and `while`) are classified as *example of* the given reserved word, thus file is associated to *EXAMPLE* class that *illustrate* concepts thus producing a link semantically relevant between the plain resource and the instructional ontology.

Semantic units: 2 items		Semantic units: 3 items		Semantic units: 4 items		Semantic units: 5 items	
Expression	Strength	Expression	Strength	Expression	Strength	Expression	Strength
Statement **while**	0.0547	Initialization finalization increment	5.0605e-003	Loop **while** for **do-while**	8.648e-0.04	Sentence **while** loop **do-while for**	3.4877e-005
Loop **for**	0.0539	Sentence repeat loop	4.4703e-003	Instruction Initialization Finalization Increment	6.4251e-004	Loop instruction Initialization Finalization Increment	1.4165e-005
Loop **do-while**	0.0495	**While** for do-while	3.4613e-003	**For** Initialization Finalization increment	4.3869e-004		
Loop **while**	0.0411	Iterate Increment loop	2.3686e-0.03	Finalization sentence loop expression	3.147e-004		
do-while	0.0372	**While** expression sentence	1.9725e-003				
Repeat	0.0320	**For** expression sentence	1.7148e-003				
Initialization sentence	0.0285	Do-while expression sentence	1.623e-003				
Termination Sentence	0.0263	Initialization sentence loop	0.9364e-003				
Initialize loop	0.0192	End Sentence loop	0.8295e-003				
Finish loop	0.0187						
Increment expression	0.0174						
Increment counter	0.0169						
Execute sentence	0.0158						
Construction loop	0.0136						
Evaluate expression	0.0114						

Table 1. Semantic units' analysis from file `text_ch5s01s01.html`

4 Conclusions

The described model provides a new approach based on unsupervised neural networks for detecting semantic units constituted by up to 5 atomic concepts that can be applied to different datasets of educational material: full-length document collections or

single documents. The characterization obtained can be used for content identification purposes including information retrieval systems, thesaurus generation or issues related to natural language generation.

The regularities of human language studied allow characterizing text documents in a better way, since this approach stresses the importance of multiword concepts instead of atomic words, and there is usually an element of meaning added to the combination of words [12] that make up a multiword. On the other hand, it holds similarities with the way human mind is supposed to work. In addition to a better semantic characterization, the study of these regularities to extract multiwords produce important savings in the number of required indexing terms to represent documents, what is traduced in savings in the amount of software memory or electronic circuitry devoted to this task.

The conclusions obtained in this work suggest that the proposed algorithm could be an interesting tool in automatic semantic units identification processes. The main inconvenience of other approaches that rely on sets of part of speech tag patterns is that noisy words that can appear between connected words avoid the multiword extraction [13], however the neural network design proposed is not subordinated to such limitation since it is not affected by words appeared between two connected. As the evaluation process shows, the system is particularly effective for detecting complex concepts made up of not only two terms but also, made up of three, four and five terms, what is particularly interesting for a better characterization of documents. The inconvenience is that the term order inside a group is lost. However a statistical process is carried out in order to solve the problem [16]. The method accuracy is rather high, the system is able to detect neuron associations due to patterns that just appear a few times in the dataset. The criterion of reciprocity used avoids that false couples provided by noise pattern (patterns resulting from chance events) could be make up, even if the selected threshold is too low.

Future work will also trend to perform more experiments using richer input like open courseware of UNED University. Also will use experience from other similar works on automatic metadata extraction and classification, as mentioned in [7] to explore non-labeled open web courses.

References

1. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press. Cambridge, Massachusetts. London, England. (1999) pp. 151-189.
2. Justeson, J., Katz, S.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering. (1995) pp. 9-27.
3. Chen, H., Lynch, K.: Automatic Construction of Networks of Concepts Characterizing Document Databases. IEEE Transactions on Systems, Man, and Cybernetics, 22. (1992) pp. 885-902.
4. Ruiz, M., Srinivasan, P.: Automatic Text Categorization Using Neural Networks. Advances in Classification Research, 8: Proceedings of the 8th ASIS SIG/CR. Ed. Efthimis Efthimiadis. Information Today, Medford, New Jersey. (1998) pp. 59-72.
5. Kohonen, T.: Self-organization and associative memory. Springer Series in Information Sciences, Vol. 8. Springer-Verlag, New York. (1984)

6. Lin, C., Chen, H.: An automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents. IEEE Transactions on Systems, Man and Cybernetics: Cybernetics, 26(1) (1996) pp. 75-88.
7. C. D. Kloos and M. Rodríguez-Artacho (2004) "Semantic Web aided Authoring of Educational Material" in proceedings of the SW-EL 06 conference
8. J. Qin & N Hernández (2004) "ontological representation of learning objects: building interoperable vocabulary and structures" in proceedings of WWW2004 conference. ACM press
9. M. Rodríguez-Artacho and F. Verdejo (2004) "Modelling Educational Content: The cognitive approach of PALO language" In journal of Educational technology and society Vol 7 (3), 2004 124-137.
10. Mohan, P. Brooks, C (2003) "Learning objects on the semantic Web". proceedings of the IEEE conference on advanced learning technologies ISBN: 0-7695-1967-9
11. MacLeod, K., Robertson, W.: A Neural Algorithm for Document Clustering. Information Processing and Management, 27 (3). (1991) pp. 337-346.
12. Firth, J. R.: A synopsis of linguistic theory 1930-1955. Studies in Linguistic Analysis. Oxford: Philological Society. (1957) pp. 1-32.
13. Smadja, F.: Retrieving collocations from text: Xtract. Computational Linguistics. (1993) pp. 143-177.
14. Caropreso, M. F., Matwin, S., Sebastiani, F. (2000). Statistical phrases in automated text categorization. Technical Report, Instituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Recerche.
15. Lin, X., Soergel, D., Marchionini, G.: A Self-Organizing Semantic Map for Information Retrieval. Proceedings of the 14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval. (1991) pp. 262-269.
16. Perez C. "Intelligent system design using artificial neural networks for automatic semantic extraction in medical literature. Second International Workshop on Intelligent System Design and Applications ISDA 2002. Atlanta, USA, August 2002.

L2Code: An Author Environment to Integrate Multiple Intelligences and Naive Bayesian Classifiers for MultiParadigm-Programming Training

M. L. Barron-Estrada, Ramon Zatarain-Cabada, J. Moises Osorio-Velazquez, L. Zepeda-Sanchez, and Carlos A. Reyes-Garcia *

Instituto Tecnologico de Culiacan, Juan de Dios Batiz s/n Col. Guadalupe, C.P. 88220
Culiacán, México, tel. +52 667 7131796
rzatarain@itculiacan.edu.mx
* Instituto Nacional de Astrofisica, Óptica y Electrónica (INAOE)
Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, 72840, México
kargaxxi@inaoep.mx

Abstract. L2Code is an Intelligent Tutoring System used for teaching programming courses for different paradigms. It is designed to work with diverse types of modules oriented to certain ways of learning using principles of Multiple Intelligences. The learning modules can be elaborated easily by any person who teaches some topic of programming. The predictive engine uses a Naive Bayes classifier which operates in real time with the knowledge of the historical performance of the student. Results of studies show that learning characteristics can be modeled using a scheme of learning with the correct election of the attributes, establishing for that reason the primary target of L2Code.

1 Introduction

Learning a programming language is in general considered a hard task, and programming courses often have high abandon rates. It has even been recognized, that it takes about 10 years for a beginner to become an expert programmer [1]. Educational research has been made to distinguish the characteristics of beginner programmers and to study the learning process and its associations to the different aspects of programming [2, 3]. Lately also differences between procedural and object-oriented education approaches have been studied, as Java and C++ have become common educational languages [4]. Milne and Rowe [5] studied in a recent survey the difficulties of Object oriented programming by performing a web-based survey for both students and teachers.

In addition, today many students study programming languages by using distance education where they can access many tutorials in the subject. The problem of this technique of learning is the lack of an actual tutor who, according to the characteristics of the student, illustrates the learning material in a comprehensible style. In order to solve the problem, much research has been conducted by proposing

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 406–416

Received 19/06/07
Accepted 31/08/07
Final version 30/09/07

using Intelligent Tutoring System (ITS) for teaching programming [6, 7, 8, 9]. However all of these works are focused on introductory programming material or are oriented to one single paradigm or programming language.

Our proposal is an ITS designed to accept diverse types of programming language paradigms oriented to different ways of learning by using the principles of Multiple Intelligences [10]. This system, named L2Code, can dynamically identify the learning characteristics of the student [11] and provide him the learning or study material according to his type of intelligence. The different programming modules can be conveniently produced by any instructor who wants to teach some programming topic. It is only necessary to specify which resources refer to which types of student intelligences, and which evaluation will be part of the different modules of the ITS. This is necessary in order to measure the student performance and to improve the prediction of the best learning resource. A predictive engine for L2Code works with a Naive Bayes classifier [12] which operates in real time with the knowledge of the historical performance of the student.

The arrangement of the paper is as follows: In Section 2, we present the architecture of L2Code describing each one of the module components. In Section 3, we discuss the implementation of several important algorithms used in the software. Some experimental results are shown in Section 4. Comparison to related work is given in section 5 and conclusions are shown in Section 6.

2 Architecture of L2Code

The general architecture of the system (figure 1) includes a set of components that allow modularization, scalability, and maintainability of the system.

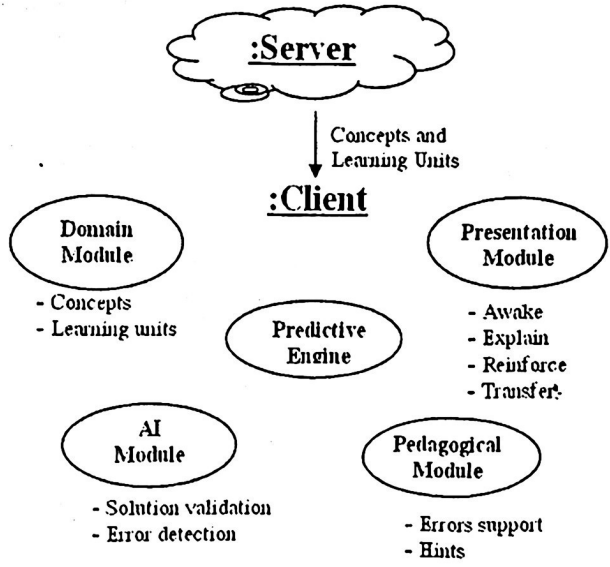


Figure 1. General architecture of L2Code

The server is the one in charge to provide the complete course that comes to be a package of different resources with its respective evaluations. The server is not more than an abstract entity, since can be distributed in internet by a Web site, or directly by the creator of the course.

The client contains the ITS. It has the following components:

- Domain Module. It is the one in charge to encapsulate the content of the course, such as concepts and learning units with their respective resources.
- Presentation Module. It is the one that works with certain unit of learning, like waking up the student, explaining some concepts, reinforcing the content or simply transferring new knowledge.
- Pedagogical Module. It is the one in charge of the tutor, making functions such as detecting errors in the answers of the student, and feed backing and guiding the student towards the correct solution.
- AI Module. Fundamental part in the operation of the pedagogical module, since it is the one that really detects the type of solution for the student, correct or incorrect, therefore the pedagogical module only worries about the feedback process.
- Predictive Engine. Its function is the one to calculate the probability that the student has taken the correct course, according to its type of intelligence measured in the degree of assimilation of the learning unit. With this calculation, the predictive engine is able to predict which would have to be the following resource that the student would have to take.

2.1 Layer Model Architecture

The layer model architecture of the system (figure 2) is defined with respect to the system scalability and maintainability. This architecture define a series of interfaces and services that are provided between each one of the layers.

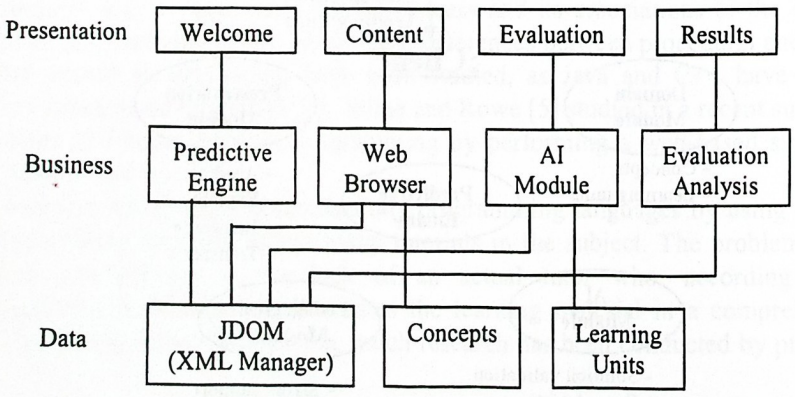


Figure 2. Layer model architecture

2.2 Learning Process in L2Code

The learning process in a module (figure 3) starts by describing basic information like name, objectives, and previous and further knowledge of the module. Next, a visualization of theoretical content is shown, and then a corresponding evaluation is performed. In this process, there exist an assistant to the student on the solution of the problems. And finally the results of the student are shown with a corresponding feedback.

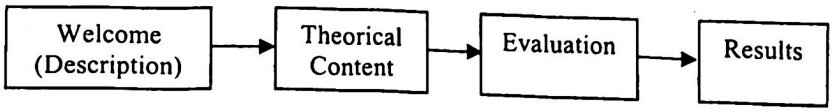


Figure 3. Learning process

2.3 Predictive Engine

As we defined previously, the predictive engine is the one in charge to calculate the probability that a student corresponds to certain type of learning resource, predicting the ideal one that the student would have to attend.

The input of the engine is composed by the results of the evaluation done to the student after the conclusion of a learning resource, and the attributes used for the evaluation, obtaining as an output the learning type of the student. This way we can indicate the appropriated resource for the student.

The following attributes have been chosen to reflect how the students use the different resources:

- Time (F, N, L). There is a range of time specified by the course creator. If the student spent less time reading the question than the low limit then *Time* is F (fast); if the time past the high limit then *Time* is L (long); otherwise *Time* is N (normal).
- First choice (Yes, No). *Yes* if the student answer is the first one he/she chose, *No* otherwise.
- Question attempted (Yes, No). *Yes* if the student attempts to answer a question, otherwise *No*.
- Accuracy (0 .. 1). Measures the approximation of the student answer with respect to the correct answer. This computation depends of the evaluation type defined by the course creator.

After determining the probability of each question, the probability corresponding to the module (resource type) is calculated considering the following attributes:

- Time (F, N, L). There is a range of time specified by the course creator. If the student spent less time reading the learning contents than the low limit then *Time* is F (fast); if the time past the high limit then *Time* is L (long); otherwise *Time* is N (normal).
- Repeat (Yes, No). *Yes* if the student had already seen this resource, *no* otherwise.
- Code value (0 .. 1). This value is defined by the course creator and says how much percentage must be assigned to code questions.

- Intelligence (VL, LM, VS, MR). It defines the type of student intelligence. According to Gardner theory [10] there are seven intelligences. We manage four of them: Verbal/Linguistic, Logical/Mathematical, Visual/Spatial and Musical/Rhythmic.

3 Implementation

The development of the system was made by following a cascade model with a modular development under the UML language [13, 14]. The system was implemented with the Java™ language [15]. L2Code makes use of two external packages that are: JDOM [16] for the XML reading and writing and SWT (Standard Widget Toolkit) [17] for the creation of natives graphical interfaces allowing all of this the multiplatform facility provided by Java™.

3.1 Naive Bayes Classifier

This algorithm (figure 4) is in charge of the probabilistic calculations to be able to make the prediction of the ideal learning resource for the student. During the interaction of the student with the learning module the attributes of this interaction are registered and, when finishing it, the corresponding probability of the actual learning resource is updated.

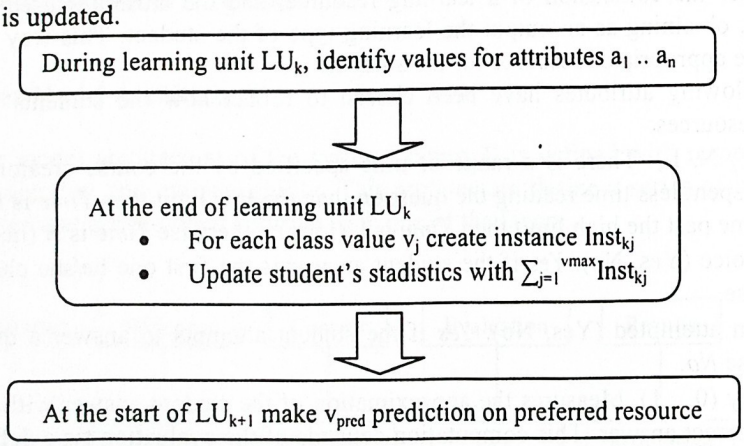


Figure 4. Naïve Bayes classifier algorithm

3.2 Evaluation Algorithms

In the process of evaluation of the learning module we define four different evaluations:

- **Multiple Options.** It offers a series of possible answers, where only one answer is correct.

- **Keywords.** Here we evaluate the answer of the student based on the amount of correct keywords that the answer contains. The algorithm is explained in figure 5.

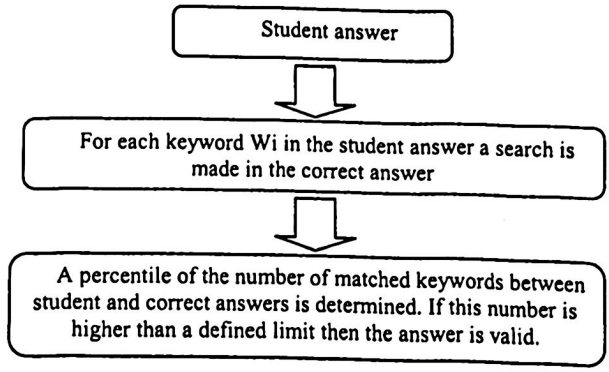


Figure 5. Evaluation with keywords

- **Edit Distance.** This is also for free answers from the student, but the evaluation method is oriented to a minimum number of characters that must be eliminated, inserted or interchanged so the answer of the student is identical to the correct answer. This is illustrated in figure 6.

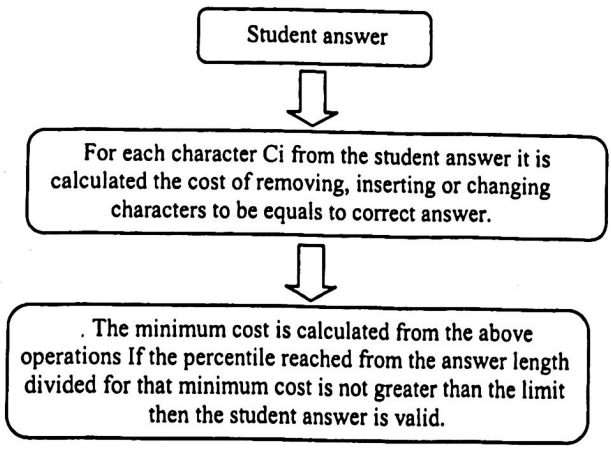


Figure 6. Evaluation with edit distance algorithm

- **Practice Evaluation (Code Problem).** This type of evaluation (see figure 7) was implemented to evaluate code and to provide hints to the student throughout its development and, at the end, a feedback of its answer.

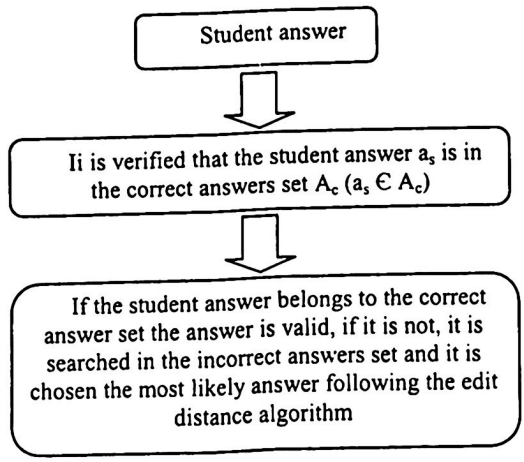


Figure 7. Algorithm for practice evaluation (code problem)

4 Experimental Results

When the student makes the election of his/her first learning module the result of the predictive engine is undecidable, since there has been no interaction with the system.

We will present an example of a student interaction with the system in a course of object-oriented programming in Java, where the student is taking the module of classes and objects.

In the Welcome section of L2Code, initial information of the learning module like objectives, description of the learning material, etc. is shown, as well as the types of presentation available.

After the student has finished the content of the module, he/she needs to be evaluated in some of the possible ways that we explained above.

When the student has finished attending one learning module and has been evaluated, a probabilistic value is determined and used for the prediction of the type of intelligence. In order to be able of comparing the final calculation with the rest of the other learning resources and to determine the appropriate resource for the student, this probabilistic value is stored and merged with the rest of the calculations made to the learning resources of the same type.

Table 1 gives an example of a student using L2Code.

Table 1. Student interaction

Student answer	Response time
methods	10
Declaration and body	35
constructor	10
True	80
"()"	20
name body arguments	80
new	25
usr = new User()	100

The interaction was in a module with Visual/Spatial intelligence type and characteristics shown in table 2.

Table 2. Module evaluation characteristics

Correct answer	Evaluation type	Normal time	Long time	Min. accuracy
method	Edit distance	15	60	80
Declaration body	Multiple options	10	60	100
constructor	Edit distance	15	60	80
False	Multiple options	10	30	100
{}	Multiple options	10	30	100
Return name	Keywords	15	60	75
new	Keywords	10	30	100
usr = new User();	Code problem	30	300	100

In table 3 we show the results of the student interaction (probabilistic calculations).

Table 3. Probabilistics results for student interaction

Accuracy	Probability
83	0.83
100	0.90
100	1.00
0	0
0	0
75	0.60
100	0.90
94	0.85

As this learning module had assigned a 20% to the practical evaluations (this is designed by the module creator), the probability that this resource has facilitated the learning to the student is of 0.65. This value later is added to the calculations done to other resources of the same type. Thus, at the beginning of another resource, the probabilities can determine that the student belongs to certain characteristics of learning.

In the end, the results of the student evaluation are shown. It is necessary to indicate that the result is different from the one used for calculating the learning type. Also the student will have the possibility of seeing his/her answers and compare them with the correct ones, in addition to receive feedback from the system.

Figure 8 presents the results for the student interaction in this running example.

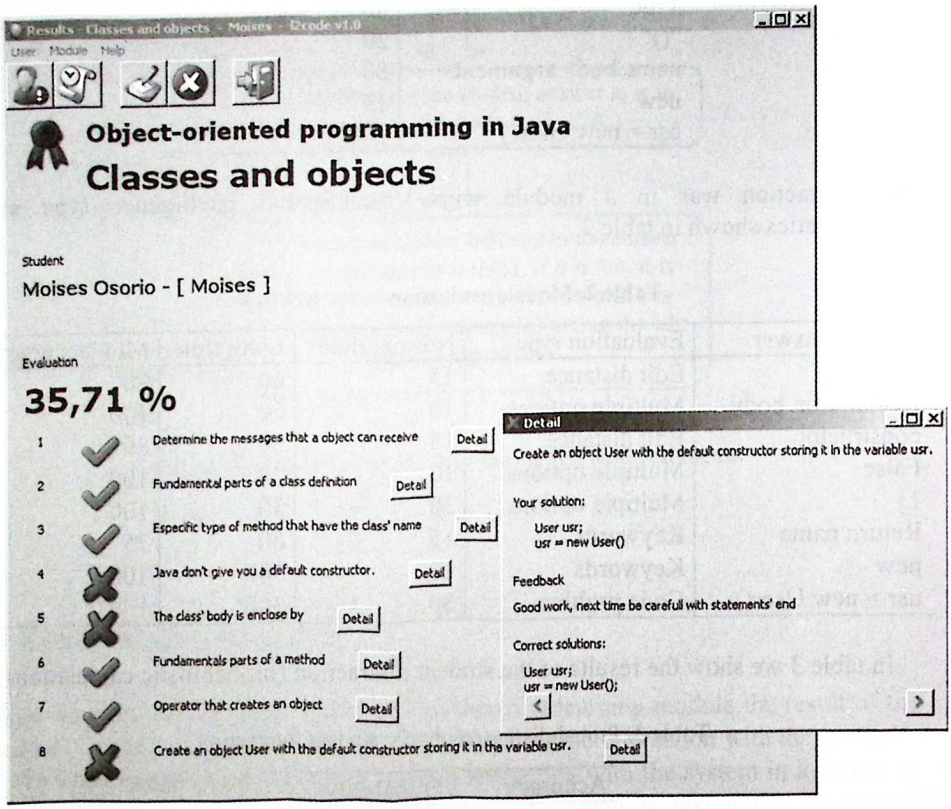


Figure 8. Evaluation results

5 Related Work

A series of research work in this area has been developed for several years; nevertheless all of them are oriented for teaching a single programming language and most of the times for introductory courses. One of them is ITEM/IP [7] an ITS for teaching programming. ITEM/IP is only oriented to provide an introductory course to Turingal (programming language). Another ITS is GREATERP [8] which is based on Anderson's theory of learning and oriented for teaching the LISP language. Last, a system named BITS [9] which is also oriented for teaching only one programming language.

As we can see, one main difference between those other systems with respect to L2Code is that they are oriented to just one programming language. Besides, the exactitude of the student answers depends of just one programming language, whereas L2Code looks for the degree of exactitude of the answer within a set of correct answers defined by the creator of the learning module. That is the main reason why L2Code does not need to specify a programming language or an exact solution but a set of answers that must be validated like correct and another set of answers that must be validated like incorrect allowing identifying the correct feedback to the student.

6 Conclusions

In this paper an ITS named L2Code was described. The system predicts the best learning resources for the students. The learning modules are a set of features that describe when the learning resource must be presented to the student. On entry to a particular unit, the learning scheme predicts which resource the student should use.

Currently some empirical studies are taking place to analyze the reaction of students to the predictive engine in L2Code. This study is examining instructional strategies due to the relationship between them and the learning performance.

Future work involves development of a user-friendly interface to create courses and further analysis in order to identify the relevance of different features. Other work will involve generalizing the adaptive engine to use different categories of learning resources.

References

1. Soloway, E. & Spohrer, J. (1989). Studying the Novice Programmer, Lawrence Erlbaum Associates, Hillsdale, New Jersey. 497 p.
2. Barr, M., Holden, S., Phillips, D. & Greening, T. (1999). An exploration of novice programming errors in an object-oriented environment, SIGCSE Bulletin, 31(4), pp. 42-46.
3. Deek, F., Kimmel, H. & McHugh, J. (1998). Pedagogical changes in the delivery of the first-course in computer science: Problem solving, then programming. Journal of Engineering Education, 87, pp. 313-320.
4. Wiedenbeck, S., Ramalingam, V., Sarasamma, S. & Corritore C. (1999). A comparison of the comprehension of object-oriented and procedural programs by novice programmers. Interacting with Computers, 11(3), pp. 255-282.
5. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B., Laxer, C., Thomas, L., Utting, I. & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students, SIGCSE Bulletin, 33(4), pp. 125-180.
6. Edward R. Sykes, Franya Franek: Web-Based Architecture of an Intelligent Tutoring System for Remote Students Learning to Program Java.
7. P. L. Brusilovsky: Intelligent Tutor, Environment and Manual for Introductory Programming

8. Brian Reiser, John Anderson, Robert Farrell: Dynamic Student Modeling in an Intelligent Tutor for LISP Programming
9. C. J. Butz, S. Hua, R. B. Maguire: A Web-Based intelligent Tutoring System for Computer Programming
10. Howard Gardner: Theory of Multiple Intelligences.
11. Declan Kelly, Brendan Tangney: Predicting Learning Characteristics in a Multiple Intelligence Based Tutoring System.
12. Markus Lang: Implementation of Naïve Bayesian Classifiers in Java.
13. Adolfo Duran, Ana Cavalcanti, Augusto Sampaio: Formal Methods and Software Enginnering
14. Robert Cecil Martin: UML for Java Programmers
15. Gosling, Joy, Steele, Bracha: The Java™ Language Specification
16. Jason Hunter, Brett McLaughlin: JDOM™ Project, http://www.jdom.org
17. Eclipse Foundation: SWT (Standard Widget Toolkit), http://www.eclipse.org/swt

Intelligent Biomedical Applications

Pre Diagnostician of Heart Arrhythmias Using Wavelet Transform and Fuzzy c Means

Ramón Miranda[1], Isaac Chairez[1], Agustín Cabrera[1]

[1] Unidad Profesional Interdisciplinaria de Biotecnología - Instituto Politécnico Nacional
Av. Acueducto de Guadalupe s/n Col. Ticoman Barrio La Laguna,
CP. 07427, México, D.F.
acabreral1@ipn.mx

Abstract. Heart pathologies are one of the main causes of death in the entire world. Making a correct pre-diagnose can be helpful in order to avoid mistaken treatments that unfortunately are still carried out. For that strong reason, it is presented in this paper, the description of a pre-diagnose process of this heart arrhythmias, using ECG signs and mathematical algorithms as the Continuous Wavelet Transform in order to characterize and identify some features of this pathologies and the Fuzzy c Means that is allowed to identify to what group of this pathologies it belonged an ECG sign of an unknown patient. This Technique could be useful in real medical processes since it can help physicians to carry out a better diagnose.

1 Introduction

1.1 Heart Arrhythmias

In recent years the index of cardiovascular illnesses has been increased, it is caused by inadequate diets, uncontrolled rhythm of life, overweight, or congenital problems among other cases. This information shows the necessity to dedicate more resources to treat and prevent these sufferings by means of improving the diagnosis systems that already exist, which are convenient only for the identification of certain heart pathologies and in spite of the fact that they make this identification with an acceptable error, it is necessary an improvement in the diagnose process.

The current methods, in the pre diagnosis of these pathologies, make an analysis based on the Fourier Transform (FT), however, one of the main inconveniences that this FT has is their non recommended use in non stationary signals. Despite, the normal heart has a non stationary signal it is stiller noted when it presents any kind of arrhythmia. Therefore, in this paper it is suggested to make a pre diagnosis using the Continuous Wavelet Transform (CWT) that has demonstrated better results in certain applications for more than ten years when the signals to be processed are in fact of the non stationary type [5], [9].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007; pp. 419–429

Received 18/06/07
Accepted 31/08/07
Final version 23/10/07

The cardiovascular illnesses are a serious problem of public health in industrialized countries, since they occupy the first place in causes of death, moreover, they are increasing the demand of medical services what generates several socioeconomic repercussions in the health area along the world. For this reason, the use and development of non invasive techniques like electrocardiography (ECG) grants a useful perspective for the diagnosis and treatment of this kind of patients. However, the signal obtained by the ECG is not enough to make a correct diagnose and it is necessary to make a signal processing in order to identify with better results whether a patient has o not any kind of heart pathology [1], [2], [6].

The intervals of time among the different waves of the ECG are important in the diagnosis because they reflect bioelectrical processes that have clinical implications when they are outside of the margin of normal variation. The custom way in which a cardiologist can classify and treat the dysfunctions of the heart rhythm, is by taking into account some ECG characteristic like heart frequency, width of the complex QRS, rhythm state, witness or absence of P waves and their relationship with the complex QRS [3].

1.2 Continuous Wavelet Transform

In the signal analysis there exists a great deal of tools that have been developed for years. Belonging to those tools is the Fourier Transform, the one that has gain considerable attention due to its capacity of represent the frequencies that are contained in a certain signal. The Wavelet analysis allows the use of big intervals of time in those segments where a bigger precision is required in low frequency, and smaller regions where information is required in high frequency [4].

A simple form of understanding the way of operation of this transform is by thinking that the signal in base of time is going to be processed by several low pass and high pass filters which allow the separation of high frequency signal portions from low frequency ones [11], [12].

The objective of the CWT is to decompose certain signals into functions denominated Wavelets components, which form a space base of functions with non finite dimension, and certain properties like orthogonal form, size, softness, duration, etc. The CWT is defined by (1) [7], [8].

$$C(a,b) = \int_{-\infty}^{\infty} f(t)\Psi_{a,b}(t)dt \tag{1}$$

There exist many families into the CWT like Haar, Coifelts and Daubechies. The last one is the used family in this work [9].

1.3 Fuzzy Logic

The human brain carries out a large amount of parallel activities to acquire information about reality, analyze it by means of previous knowledge and to produce an adequate response to the corresponding external excitation. All these processes are de-

veloped without mathematical modeling on surrounding systems. This reasoning is main principle of fuzzy logic [10].

In fact, this mathematical theory was developed to reproduce, as far as possible, the human brain capability in order to analyze the real life. The fuzzy logic works with the so-called membership function (MF) as main element to reproduce the uncertainty characteristic on human reasoning. This (MF) concept allows constructing a class of conditions that related the actual particular characteristic on certain process and then to supply a consequent interpretation.

The fuzzy logic methodology introduces the fuzzy groups that relate some particular subsets (MF) defined inside a certain universe of numeric values (representing the real variables involved in the process under study) with some linguistic variables (that are related with the physical meaning of variables nature). This process is known as Fuzzification method, where the real values are evaluated by the MF mathematical representation.

This class of function evaluation is quiet different from the classical Boolean logic, because here the membership degree belong to the closed set (0, 1). Once this process has been completed, a particular methodology is performed to create a conclusion representing an adequate solution to the current situation provided by the actual input variables. Finally, these mathematical expressions are transformed into real physical variables in the Defuzzyfication process.

2 Methodology

In this section it is presented the description of the pre diagnostician process, starting from the data base was used in this work, the signal characterization by the CWT and the clustering by Fuzzy c-Means (FcM).

2.1 Data Base

The database was taken as reference was the MIT - BIH Arrhythmia Dates Base that Contains 48 fragments, corresponding to ambulatory registrations of two channels. The sampling frequency is of 360Hz, with 11 bits of resolution and a range of 10mV. The following table summarizes the characteristics of the used registrations in this work.

Register	*Gender*	*Age*	*Signal Characterization*
101	Female	75	Normal Rythm
104	Female	66	PVC Pacermarker Fusion
105	Female	73	Normal and PVC
107	Male	63	PVC, Paced
109	Masculino	64	Left BBB y PVC
112	Male	54	Normal and APC
200	Masculino	64	High frequency ventricular beat

Table 1. Selected signals of the MIT – BIH arrhythmia data based.

2.2 Mathematical Description of the Wavelet Algorithm

ECG signals obtained by the MIT data base was introduced into a computer. The vector function that represents the bioelectric heart signal is described by (2).

$$\tilde{f} := (x_1, x_2, \ldots, x_n) \tag{2}$$

Every ECG signal is divided into the same number of parts, starting from previous knowledge the size of this intervals was selected as 1 second of acquired signal. The mathematical representation of these intervals is described by (3).

$$\tilde{f} := (\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_k) \tag{3}$$

Every interval is composed by point series (4).

$$\tilde{f}_i := (x_{1i}, x_{2i}, \ldots, x_{\lambda i}) \tag{4}$$

This points are used in order to form a matrix like is shown in (5).

$$F = \begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{\lambda,1} & 0 \\ \vdots & & & & \vdots \\ x_{1,k} & x_{2,k} & \cdots & & x_{\lambda,k} \end{bmatrix} = \begin{bmatrix} \dot{f}_1 \\ \vdots \\ \dot{f}_k \end{bmatrix} \tag{5}$$

The CWT is applied in every $\dot{f}_i$ and the obtained transform is the expression (6).

$$W_{j,k} = \sum_{\lambda i \in Z} \dot{f}_i[\lambda i] \Psi_{j,k}[\lambda i] \tag{6}$$

Which is a new matrix W_i, shown in (7), that contains the correlated coefficients $c_{f,\lambda k}$ of the corresponding vector $\dot{f}_i$.

$$W_i = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,\lambda 1} \\ \vdots & & \ddots & \vdots \\ c_{f,1} & c_{f,2} & \cdots & c_{f,\lambda k} \end{bmatrix} \tag{7}$$

With the aim of avoid obtain negative coefficients in the process of getting the corresponding energy, it was necessary to obtain the absolute values of the matrix like is described in (8).

$$\hat{W}_i = \left| W_i\{\dot{f}_i\} \right| \tag{8}$$

Finally, it is obtained the proper value of energy of each matrix with the purpose of classify every divided part in frequency and energy. The way was obtained the CWT energy is described in (9).

$$\upsilon_i = \frac{1}{r} \sum_{a=1}^{j} \sum_{b=1}^{r} \hat{W}_i(a,b) \tag{9}$$

Where a is the number of rows and b ius the number of columns.

Once the energy and frequency of every part of the complete ECG signal was obtained, it is possible to visualize different clusters related with every patient condition.

2.3 Mathematical Description of the Fuzzy c-Means Algorithm

The Fuzzy c-Means technique is one of the mostly used clustering tool that has gain considerable attention in real practical applications. This FcM algorithm was developed using an optimization technique in order to solve a minimization problem of a function described by (10).

$$\min_{u_f \times R^{nc}} \left\{ J_m(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{M}\sum_{j=1}^{c}(u_{ij})^m \left\|\mathbf{x}_i - \mathbf{v}_j\right\|^2 \right\} \tag{10}$$

Were $1 < m < \infty$, $V = [\mathbf{v}_1, \mathbf{v}_2 ..., \mathbf{v}_c] \in R^{nc}$ and $u_{ij} = u_j(\mathbf{x}_i)$ is the MF which assigns $\mathbf{x}_i$ to the j cluster, and the matrix $M \times c$ established by $\mathbf{U} = [u_{ij}]$ is a partition of FcM given by U_f, which is defined by (11).

$$U_f = \left\{ U \in R^{Mc} u_{ij} \in [0,1],\ \forall i, j;\ \sum_{j=1}^{c} u_{ij} = 1,\ \forall i;\ 0 < \sum_{i=1}^{M} u_{ij} < M,\ \forall j \right\} \tag{11}$$

Then it is defined the group $\Gamma_i = \left\{ j 1 \le j \le c : \left\|\mathbf{x}_i - \mathbf{v}_j\right\|^2 = 0 \right\}$ and its complements $\tilde{\Gamma}_i = \{1,2,...,c\} - \Gamma_i$. The cardinality Γ_i of the group Γ_i represents the number of prototypes that coincides with $\mathbf{x}_i$. The necessary conditions to solve $(\mathbf{U},\mathbf{V}) \in U_f \times R^{nc}$ of the minimizing function are shown in (12) if $\Gamma_i \neq \varnothing$, then $u_{ij} = 0$, $\forall j \in \Gamma_i$, y $\sum_{j \in \Gamma_i}$, $u_{ij} = 1$, $\forall i$ or (13) if $\Gamma_i = \varnothing$.

$$u_{ij} = \left(\sum_{\zeta=1}^{c} \left(\frac{\left\|\mathbf{x}_i - \mathbf{v}_j\right\|^2}{\left\|\mathbf{x}_i - \mathbf{v}_\zeta\right\|^2} \right)^{\frac{1}{m-1}} \right)^{-1},\quad 1 \le i \le M;\quad 1 \le j \le c \tag{12}$$

$$\mathbf{v}_j = \frac{\sum_{i=1}^{M}(u_{ij})^m \mathbf{x}_i}{\sum_{i=1}^{M}(u_{ij})^m},\quad 1 \le j \le c \tag{13}$$

The clusters Fuzyfication produced by the FcM algorithm is controlled by the $m > 1$ parameter.

3 Graphical and Numerical Results

In this section it is presented the obtained graphical and numerical results. Firstly it is presented the ECG signal obtained by the MIT data base, then the graphical results of the CWT application, the relation among frequency and energy and finally the clustering by FcM. Making use of the signals described in the table 1, we analyze every one with this technique. In figure 1, it is presented the signal of a female 75 years old patient with normal ECG signal.

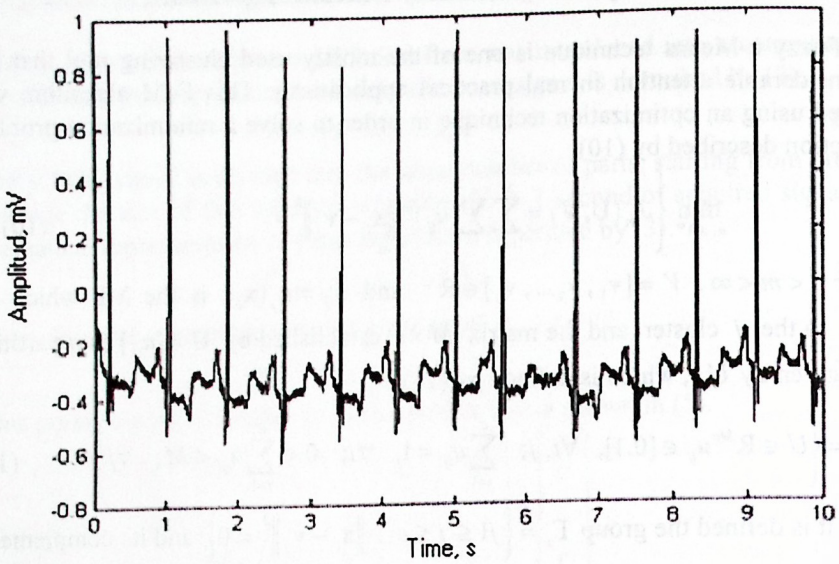


Fig. 1. 100 ECG signal segment.

Once it is applied the CWT the coefficients obtained are 3D plotted and shown in figure 2.

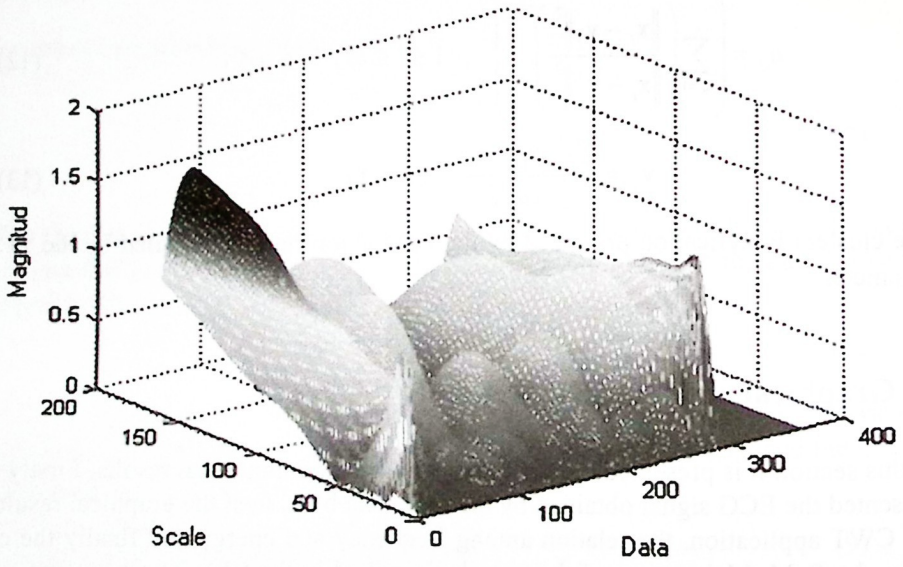


Fig. 2. Coefficients of the 100 ECG signal.

Another ECG signal belonged to a 64 years old male patient with left BBB and PVC condition. This signal is stored in the 109 reference number in the MIT data base. The ECG signal is shown in figure 3.

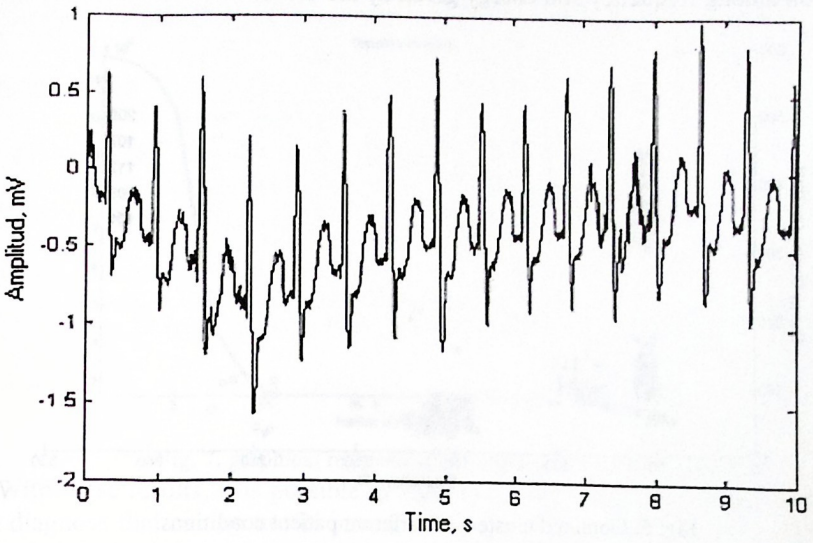


Fig. 3. 109 ECG signal segment.

The CWT application to this signal gives the coefficients that are shown in figure 5.

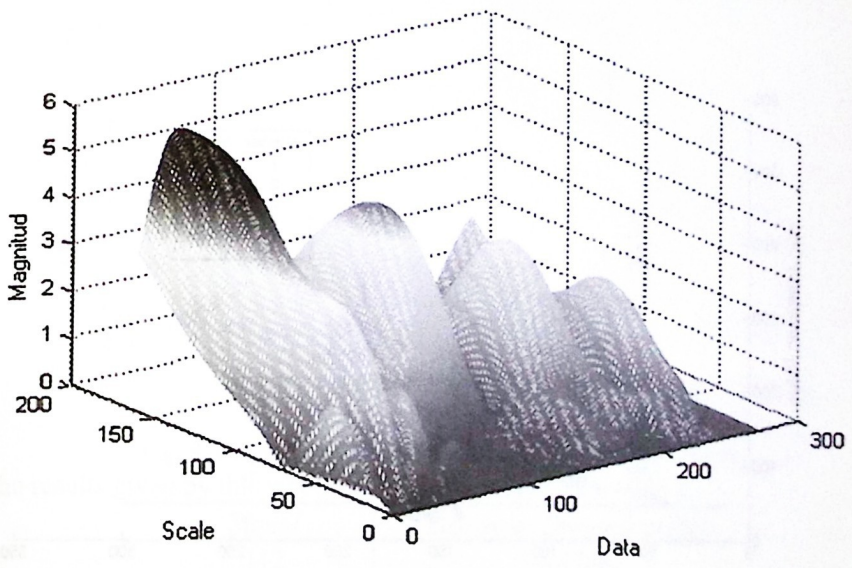


Fig. 4. Coefficients of the 109 ECG signal.

Other signals had similar way. In figure 5, it is shown the obtained clusters given by different ECG signals belonged to diverse patient heart conditions and it can be

seen that this algorithm is capable to identify different heart pathologies starting from the relation among frequency and energy given by the CWT.

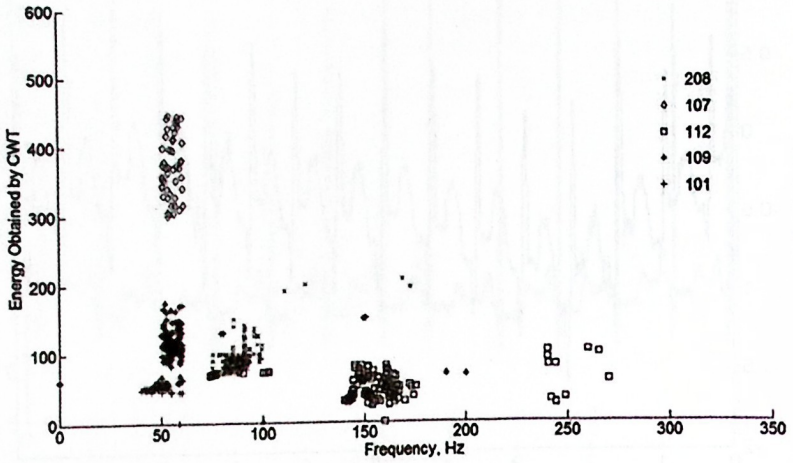


Fig. 5. Obtained clusters of different patient conditions.

Once it is obtained some clusters, the FcM algorithm is applied in order to identify to what group of heart condition is belonged an ECG sign. The obtained results of identification are shown in figure 6 where the continuous lines represent the trajectories of every center given by the FcM technique.

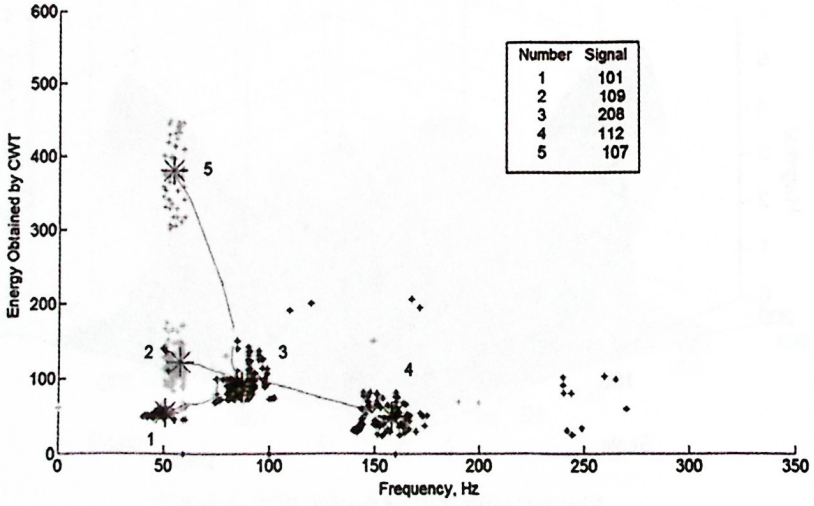


Fig. 6. Identification of heart condition by Fuzzy c-Means.

It is possible to obtain a graphical representation of the minimization problem solution plotting every iteration with its proper value. This evolution is shown in figure 7.

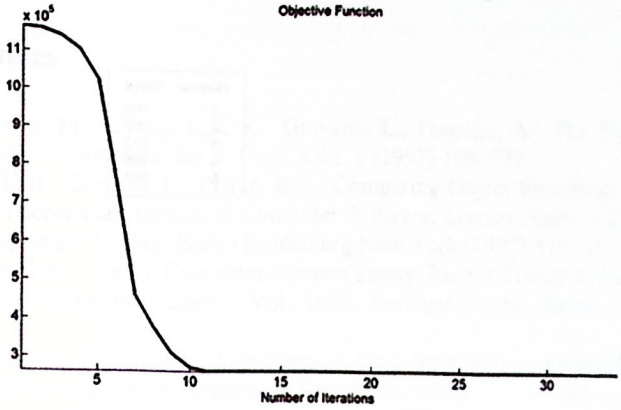


Fig. 7. graphical representation of the minimization solution.

With these results, it is possible to identify an unknown ECG signal and to get a pre diagnose that can be useful in real medical processes. In figure 8 it is presented the relation about frequency and energy of an ECG unknown condition signal and the centers of the characterized known signal given by the FcM technique.

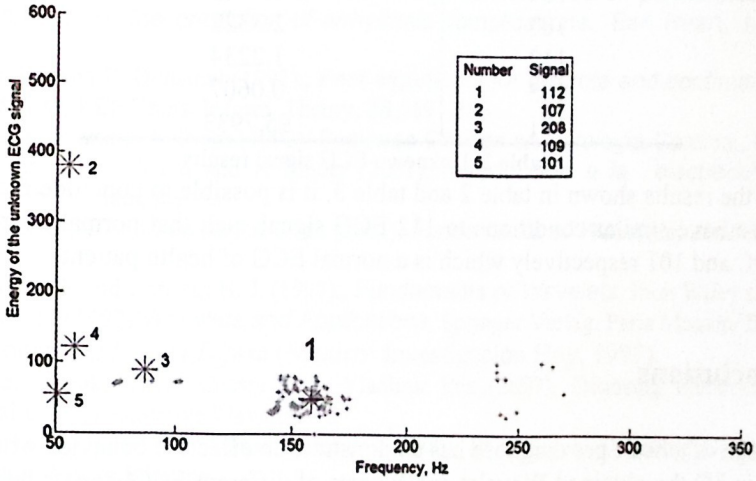


Fig. 8. Unknown ECG signal and the FcM clustering centers.

The results given by this whole technique is presented in table 2.

Signal	*Degree of Ownership [%]*
112	76.4748
107	0.5617
105	12.9137
104	3.7815
101	6.2683

Table 2. Unknown ECG signal results.

Another example of this application is related with the figure 9, where it is presented another different ECG signal of an unknown condition of heart pathologies.

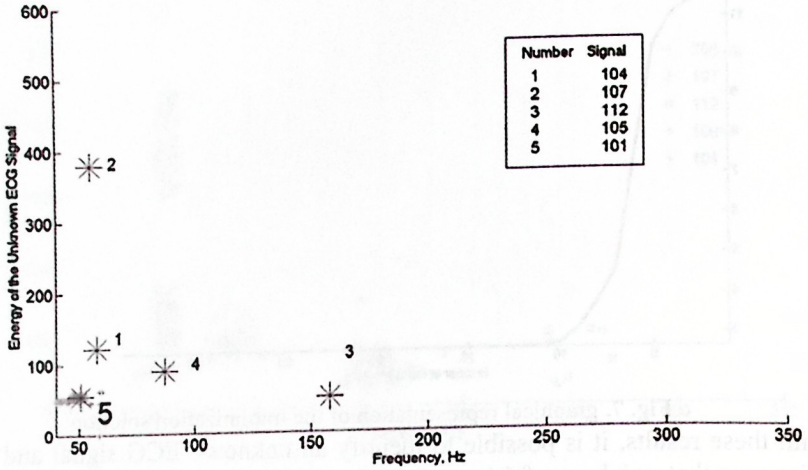


Fig. 8. Unknown ECG signal and the FcM clustering centers.

The obtained results when the FcM algorithm is applied are presented in table 3.

Signal	*Degree of Ownership [%]*
104	0.5903
107	2.3323
112	1.2234
105	0.0607
101	95.7933

Table 2. Unknown ECG signal results.

With the results shown in table 2 and table 3, it is possible to conclude that the patient most have similar conditions to 112 ECG signal such that normal heart rhythm with APC and 101 respectively which is a normal ECG of health patient.

4 Conclusions

The design of a heart pre diagnose has demonstrate an effective behavior while representing in 3D the obtained Wavelet coefficients of different ECG signals belonged to normal and with arrhythmia conditions. It is possible to achieve distinguishing high frequencies and corresponding energies of every signal. With the use of 3D plots obtained with the Continuous Wavelet Transform it is possible to localize crests that corresponds to every wave located in the scale - time plane, furthermore it can be observed that smaller scales correspond to higher frequencies such that frequency diminish when scale grows. Once the CWT is applied, the signal can be represented in 2D frequency - energy plane making clusters depending on the patient condition with the result that the pathologies studied here can be characterized with a representative center using the Fuzzy c-Means technique. In equal manner, this algorithm is

suggested to be used in medical applications to analyze different ECG unknown signals in order to help physicians to make a better diagnose.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
3. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438
3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)

1. Hsiao-Lung Chan, Shih Chin, Yu-Lin Ko, *Heart Rate Variability Characterization in Daily Physical Activities Using Wavelet Analysis and Multilayer Fuzzy Activity Culstering*, IEEE Transaction on Biomedical Engineerimg, Vol 53, No 1, January 2006
2. Somons, M.B. (1981). Use of signals in the terminal QRS complex to identify patients whit ventricular tachycardia after myocardial infraction. *Circulation* **64**,235-242.
3. Malik, M., O. Odenuyiwa, J. Poloniecki, P. Kulakowski, T. Farrell, A. Stauton and A. J. Camm (1991). *Late potentials after acute myocardial infraction performace of different criteria for the prediction of arrhythmic complications., Eur. Heart. J. 13, 599-607.*
4. Rioul, O. and P. Duhamel (1992)., *Fast algorithms for discrete and continuos wavelet transform*,IEEE Trans. Inform. Theory, **38**,569 - 586.
5. Navarro, F. and F. L. Plaza (1992). *Seciones Clinicas en Patologia Cardica.*, Meditor.
6. Dale, B., U. Dubin and K Linder (2001), *Introducción a la electrocardiografia*, Springer - Verlana Iberica.
7. Jaideva, C., G. Andrew and K Chan (2001). *Fundaments of Wavelets, Jhon Wiley and Sons.*
8. Hartmut, D. and Hartmut H. J. (1997)., *Fundaments of Wavelets*, Jhon Wiley and Sons.
9. Meyer, Y. (1992).*Wavelets and Applications.* Springer Verlag. Paris Masson/ Berlin.
10. V. Ponce, *La Lógica Difusa* (Mexico: Investigación Hoy, 1997).
11. Daniel Novak, David Cuesta Frau, Vladimir Eck (200?); Denosing Electrocardiogram Signal Using Adaptative Wavelets.
12. Nikolay Nikolaev and Atanas Gotchev (200?), ECG Signal Denosing Using Wavelet Domain Wiener Filtering.

Hierarchal Fuzzy Logic Medical Database with Decision Algorithm for Metabolic Syndrome Diagnosis

I. Salgado [(1)], A. Rodríguez [(1)], J. Chairez [(1)], A. Zúñiga [(2)] and P. Santillan [(2)]

[(1)] Unidad Profesional Interdisciplinaria de Biotecnología del IPN.

[(2)] Instituto de Ciencias Médicas y Nutrición "Salvador Zubirán"

jchairez@ctrl.cinvestav.mx

Abstract. Probably, the most complicated aspect in medical care is an efficient and opportune diagnostic from the Medic. Few years ago this branch of medical attention was depending only on physician's experience and his knowledge, who were dedicated to the patients care. However, this practice has been changed since Data Base (DB) related and intelligent systems (like experts systems) have been used. In this document a Related DB for Nutriology Clinician Department of Instituto Nacional de Ciencias Médicas y Nutrición (INCMNSZ), is shown, whose objective is to classify the patient's information for a better clinical state visualization to be gotten. An appropriate designation for an operational risk scale that indicates the attention degree that must be applied in sick persons will be determinate by the mixed application of fuzzy logic and relational database. The information system response is based in a technique called Fussy Logic, which is applied in medical variables analysis in order to find the existence of Metabolic Syndrome in the patient, and in a direct test about their life quality, alimentation habits and Clinic History. The connection between the DB and Fussy System was created with Matlab and the ODBC Data Transfer System.

1 Introduction

The clinical record is the most important element in the medical treatment that can be applied to a patient by a physician, due the information included into it. Roughly speaking, the data set contained in the corresponding patient file, is constructed by the answers (given by a specific patient) to specific question designed by an expert medical group. The questions are related to the patient's health conditions, the sickness events over the entire life, a short version of the patient's family medical record, lifestyle, some relevant accidents, surgery interventions, allergies, vaccination history and any current medicine intake by the patient [1].

These groups of data are one of the most important aspects for the medic to take decisions in a correct way, diagnose efficiently the illness' patient and propose a solution to him. In this paper the CH is a key to follow the patients who are candidates to suffer *Metabolic Syndrome*. The metabolic Syndrome is defined by the OMS as an altered regulation of glucose or diabetes (it means an insulin unresponse

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 430–441

Received 20/06/07
Accepted 31/08/07
Final version 22/10/07

that is defined as glucose make out under the last quartile levels for population in the study), besides 2 or more of the following components:

- High blood pressure.
- A triglyceride level above 150 mg/dl
- A High density lipoprotein level (HDL)
- Obesity or high IMC.

The exact cause of metabolic syndrome is unknown. Most researchers believe it is caused by a combination of the genetic makeup and lifestyle choices, including the types of eaten food and the level of physical activity. If the metabolic syndrome is diagnose, the body suffers a series of biochemical changes. Over time, these changes lead to the development of one or more associated medical conditions. The sequence begins when insulin, a hormone excreted from your pancreas, loses its ability to make your body's cells absorb glucose from the blood-your body uses glucose for energy. When this happens, glucose levels remain high after you eat. Your pancreas, sensing a high glucose level in your blood, continues to excrete insulin. Loss of insulin production may be genetic or secondary to high fat levels with fatty deposits in the pancreas [2]. Until today, the SM's diagnose is given for the Nutriology medic's experience. Actually an automatic or semiautomatic system, that can give us an appropriate and exactly diagnose, doesn't exist because each factor of SM has a dissimilar relative weigh and affect in a different way the metabolic patients state, which can't be so solved whit classic programming methods.

The Fuzzy Logic (FL) can associate uncertainly o eventually variables whit a mathematic function, which is called Membership Function, obtaining an associate grade inside a range between 0 and 1; that is one of the differences whit the classical binary theories, because the only can give two different values to the variables. The FL is conformed for four steps: Fuzzification, rules base, inference mechanism and defuzzification [3]. The FL is used in a lot of applications, like an uncertainly models controller, expert systems and decision systems whit partially defined variables. This is making whit an input's review, based in the *modus-ponens* structure: *condition-action.* An example applied to this work is the following:

If ... the patient has a high glucose level... so the patient has diabetes

These kinds of sentences are going to use to determinate SM and its related illness.

2 Methodology

The Methodology is described for the following steps:

2.1 Design of the Nutriology Department's Data Base

Microsoft Access was used to design de Data Base because it has an easy form to development the elements (tables, forms, queries, etc) of it. MatLab and the OBDC Data System were the tools to export and import data from the Data Base to the fuzzy system, which were done in Matlab. The DB has eight areas: *I.* Diagnose, *II.* Clinician

Information, *III.* Life Style, *IV.* Poverty, *V.* Habits, *VI.* Metabolic Syndrome, *VII.* ECD and Comorbility, *VIII.* Nutritional Evaluation.

2.2 Data export and import

Matlab has a different kind of tools to link it whit external software, in other words matlab can import data from access to make an analysis of the DB. "*Visual Query Builder*" is software to take the information inside the DB tables and show it in a matrix form in MatLab.
The program code to import de information from the DB is the as follows:

```
1. logintimeout(10);
2. conn = database('bd2', '', '');
3. ping(conn);
4. curs = exec(conn, 'select * from prueba');
5. setdbprefs('DataReturnFormat','numeric');
6. curs = fetch(curs);
7. z = curs.Data;
```

Description of the code Line1. Specify the time connection to Matlab before send an error message. Line 2 and 3. "database" is the function to make the link between the DB and Matlab, the function contains the DB's name, user name and password The line 3 specifies the connection state. Line 4, 5, 6 and 7. The fields in a table are chosen whit the function shown in line 4. The Data Format Return are given by the function setdbprefs.
The program code to export information from Matlab to the DB is the following,:

```
1. setdbprefs('DataReturnFormat','cellarray');
2. d='MALA';
3. exdata={d};
4. colnames(i) = {'calvida'};
5. insert(conn, Prueba, Colnames, exdata)
```

Line 1. Specify the format to return the data. Line 2 and 3. These lines show the data thal wll be export to access. Line 4. Shows the name of the DB field. Line5. To export the information to access is used the insert function which inserts a new line into the DB.

2.3 Pre- Diagnostic fussy System Development

A specialist medic and a lot of bibliography sources were consulted to make a correct diagnose, in special the National Cardiology journal was read and the diagnose algorithm was taken from this journal. This algorithm is shown in the picture 1 and describes the steps to follow to find evidence of SM. The diagnose standards are explained in the figure 1. The fuzzy system only analyzed the part in bold of the algorithm in figure 1, the variables to get involved used the parameters shows in table 1. The figure 1 shows the diagnostic algorithm. The variables included in the MS description and its corresponding range is depicted in Table 1. The parameters defining the discourse universe are shown in the same table.

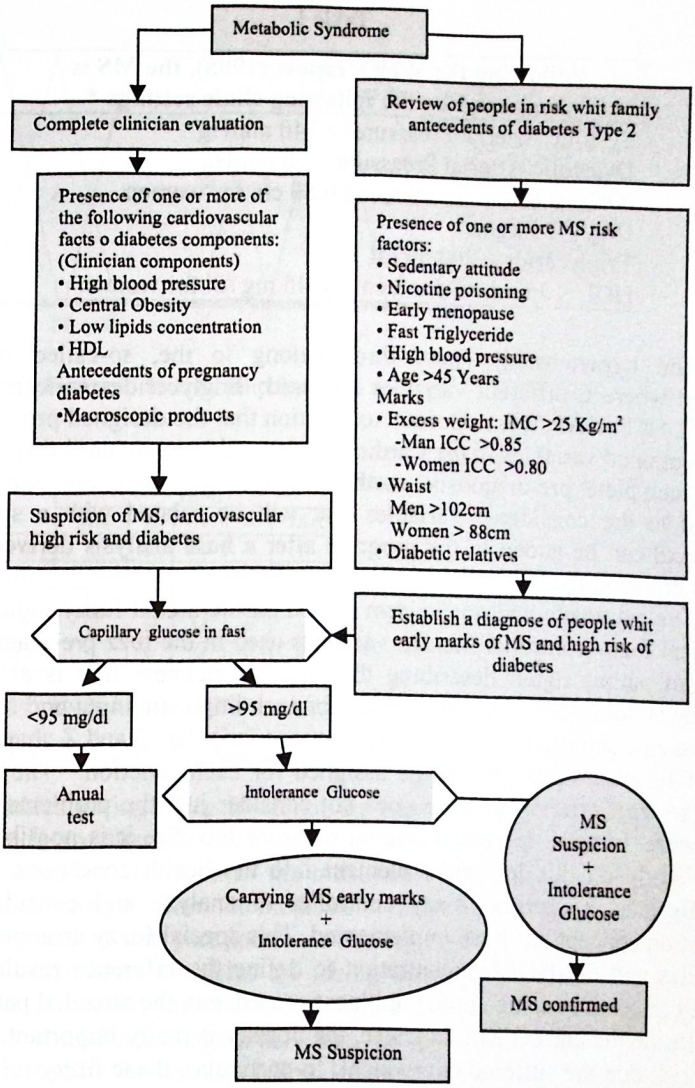


Figure 1. Diagnose algorithm by

These are diagnostic criterion: It is considered that an abnormal fasting condition and with high glucose concentration, glucose intolerant or affected by 2-type diabetes mellitus had MS if and only if there is any possible association with both components enlisted above.

FUZZY ALGORITHM DEVELOPEMENT

There were designed 3 different fuzzy pre-diagnostic computer programs. Each one of them is dedicated to one of the following clinical conditions: low lipids concentration,

Table 1

Following the WHO report (1998), the MS is componed by the following clinic settings *
Systolic Arterial Preassure ≥ 140 mmHg
Dyastolic Arterial Preassure ≥ 90 mmHg
ICC ≥ 0.85 cm for men and 0.80 cm for women
IMC ≥ 25 kg/m^2
Triglycerids ≥ 200 mg/dl
HDL < 35 mg /dl for men and 45 mg /dl for women

obesity and hypertension. These three belong to the, so-called overall clinic evaluation where 6 different variables are used: Triglycerides concentration, HDL, IMC, ICC, SAP, DAP. It is important to mention that the designed program don't use all the mentioned variables in the Cardiology Journal because until this moment there is not a complete pre-diagnostic method. However, there are another illnesses associated to the considered variables that will be related with a specific health problem and can be added to the program after a hard analysis derived by medical protocols driven at INCMNSZ.

The automatic decision system (based on hierarchal fuzzy logic) is based on the classical fuzzy controller, i.e. the variables used in the fuzz pre-diagnostic system includes an output signal describing the possible sickness that is affecting to the patient, the membership function selected for each linguistic input and output variable (for the sake of simplicity, they were chosen as triangular, S and Z shapes as usual in mostly of fuzzy designs), the range assigned for each function. The fuzzy system designed to treat arterial pressure does not consider just the problems associated to high pressure, but the decreased arterial pressure too. So, it is possible to define a complete study on this important element into the health conditions given for the patient. Besides, a specific fuzzy controller to analyze and consider low lipids concentration concept has been implemented. This special fuzzy arrangement uses the triglycerides and the HDL concentration to define the inference result, and then to provide a correct diagnostic about the illness presence in the attended patient.

Inside the correct MS diagnose, the obesity is really important to specify the patient's risk due the suffered overweight. In particular, these fuzzy relationships are based on two simple concepts: a) the relation or the, so-called, hip-waist index and b) the corporal mass index. These both concepts are so useful to define in a unique form the obesity condition on the patient. An example of the fuzzy system process to diagnose one of the three illnesses mentioned (***Low Lipid Concentration***) is shown by the following membership functions (Figure 2) and its corresponding knowledge fuzzy matrix (Table 2).

The fuzzy matrix to compare both clinician aspects gives us de rules to diagnose the illness

The membership functions (Input and Output) and the matrix were suggested by the nutrition specialist at hospital after a detail revision of the article in the Cardiology Journal. The defuzification was made by the centroid method (equation 2).

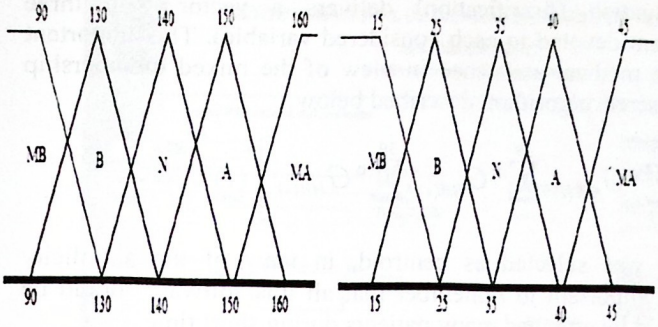


Figure 2. Membership functions for triglyceride and HDL

Low Lipid Concentration		HDL				
		MB	B	N	A	MA
Triglycerides	MB	P	P	N'	N	N
	B	P	P	N	N	N
	N	P	N	N	N	N
	A	V	V	V	P	P
	MA	V	V	V	V	P

Table 2. Knowledge Fuzzy Matrix

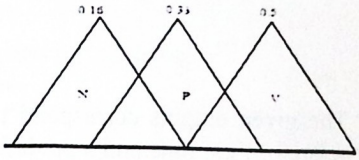


Figure 3. Output Membership Functions

2.4 Designation for an Operational Risk Scale Fuzzy System

The main variables taking place to assign the risky condition for the patient and to determinate the future attention level with the corresponding self-care are linked whit the MS diagnostic, but also include the meal habits, the carcinogenic tumors and a complete clinician analysis from the medicals to the patients. The next table illustrates the method followed in this part and the variables ranges are also depicted there. In this paper, just 4 different variables are taking into account to define the necessary medic attention. However, the real DB designed considers 39 different elements to produce a more confidence diagnostic. Once the variables are extracted of the DB, they are fuzzificated by the corresponding designed membership functions (Figure 4). The "risk" functions, for example, are classified in 1) minimum, 2) reasonable and 3) high patient risk. In general, this fuzzy structure is applied in all others variables.

Variable	Minimum Risk	Reasonable risk	High Risk
Glucose	<90	<150	>300
Triglycerides	<150	<300	>500
Uric Acid	<6	>8	<3
Albumin	>3.5	>3.4	<2.9

Table 5.MS Variables

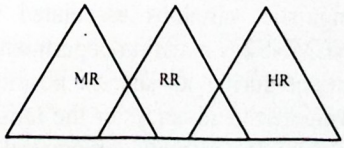


Figure 4. Input Membership fuctions

The ultimate evaluation (fuzzification) delivers a vector with three components (each one of them devoted to each considered variable). This important element, in fact, defines the medical assistance in view of the mixed membership function describing by the discrete algorithm, described below:

$$^{\circ}G = \left[\sum_{i=1}^{39} {}^{\circ}G_{MR(i)}, \sum_{i=1}^{39} {}^{\circ}G_{RR(i)}, \sum_{i=1}^{39} {}^{\circ}G_{HR(i)} \right] \quad (1)$$

The fuzzification method was selected as centroid, in view of the simplicity demanded for this job. It is important to remember that all this software should be working 24 hours and it should be attended many patients during short time.

$$y_q^{crisp} = \frac{\sum_{i=1}^{R} b_i^q \int_{Y_q} \mu_{B_q^i}(y_q) dy_q}{\sum_{i=1}^{R} \int_{Y_q} \mu_{B_q^i}(y_q) dy_q} \quad (2)$$

The given outputs correspond to the attention model: a) First time visiting the physician, b) First time and a predefined consequent attention and 3) Hard Nutritional Support (Figure 5).

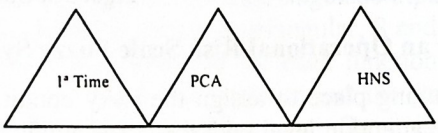


Figure 5. Membership function output

3 Results

At this moment, the DB is receiving the patient's information, however, many designs related with it have been developed: for example the visual interfaces and the corresponding formularies. The final DB design was completed using the commercial software ACCESS and using the ODBC transferred method. Additionally, this database contains some special request that allows to assign numerical values for each linguistic variables associated with the original MS scheme proposed by the INCMNSZ's nutrition department (Fig. 6). Besides, the DB was build to attend each patient during its last six hospital appointments; this fact permits the physician to evaluate the advances or the falling backs in the patient's health. This aspect is novel because the current commercial databases just give a general view on the patient response but they do not suggest any kind of treatment and, obviously, many of them are not provided by an artificial intelligence algorithm to define the corresponding diagnostic.

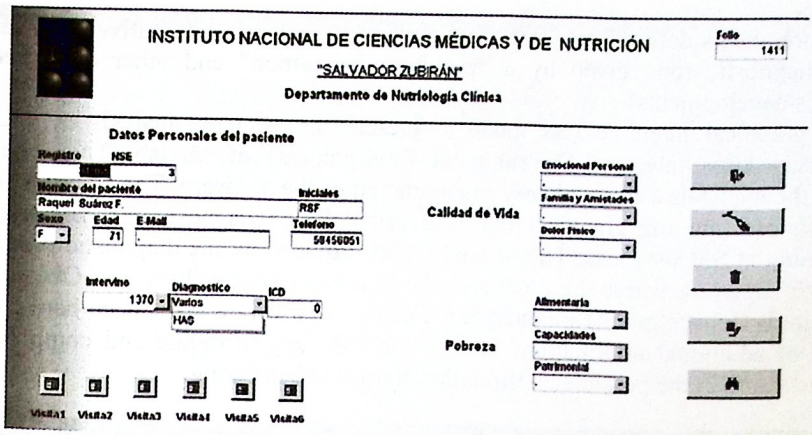


Figure 6. Lifestyle and general data for the treated patient.

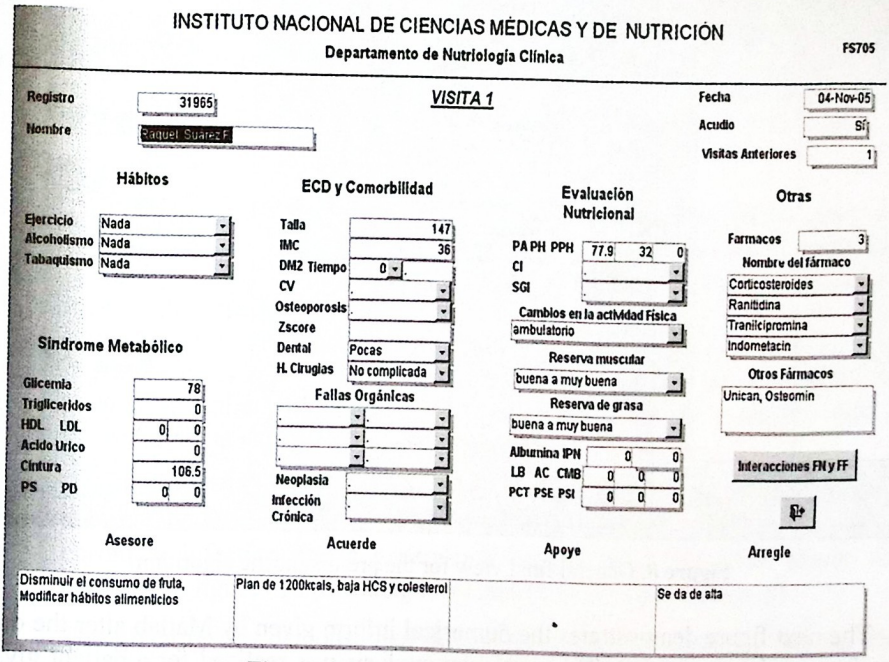


Figure 7. Clinical variables for each patient.

The figure 6 shows the DB view corresponding to the general data for each patient as well as those related with the lifestyle carried out by him or her. The next figure 7 demonstrates the frequently questions supplied by the medic to the patient and his/her clinical studies that were suggested before. Of course, these elements are used as the input data for the decision systems generated by the fuzzy arrangement.

The DB was linked with MATLAB in order to analyze the stored data. It is important to note the numerical representation considered at this point because this

procedure allows defining any statistical method to derive an alternative form for the same diagnostic (one given by a "possibility condition" and other described by statistics developments).

A graphical interface was made to present the numeric analysis by GUIDE (Graphical User Interface Development Environment) of Matlab. This interface allows the medicals a fast and easy manipulation of the software. The figure 8 shows the diagnose into the graphical interface called DIS (Diagnose Interface System) concerning at SM diagnose. The official report provided by the diagnostic process is shown in figure 8, where the three specific illnesses (Arterial Pressure, Obesity and Low Lipids Concentration) are analyzed. Finally the software introduced here gives a couple of additional information: 1) a possible therapy to define and complete the diagnostic and 2) the possible confirmation for the MS suffering.

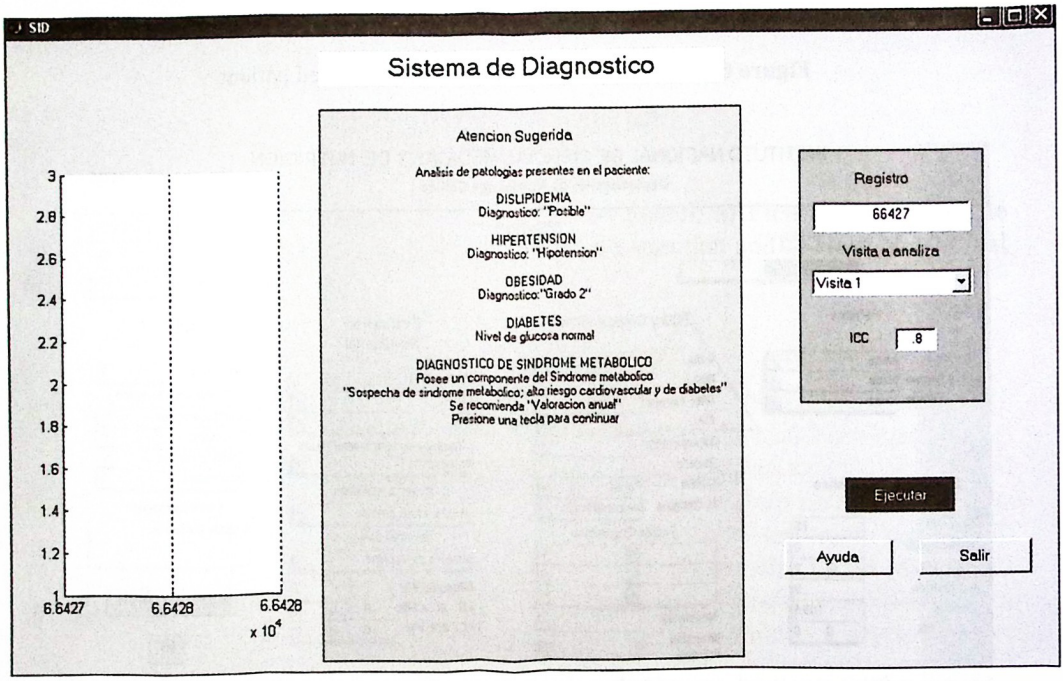


Figure 8. General final view for the pre-diagnostic algorithm.

The next figure demonstrates the numerical inform given by Matlab alter the data were imported from the DB. This particular analysis was realized for a patient given in the previous diagnosis analysis (Figure 8).

- Sex: Female. This patient is affected by the MS illness and by the type II Diabetes Mellitus (bad glucose regulation), high HDL, uric acid outside of the normal ranges and obesity (classified as level 1).

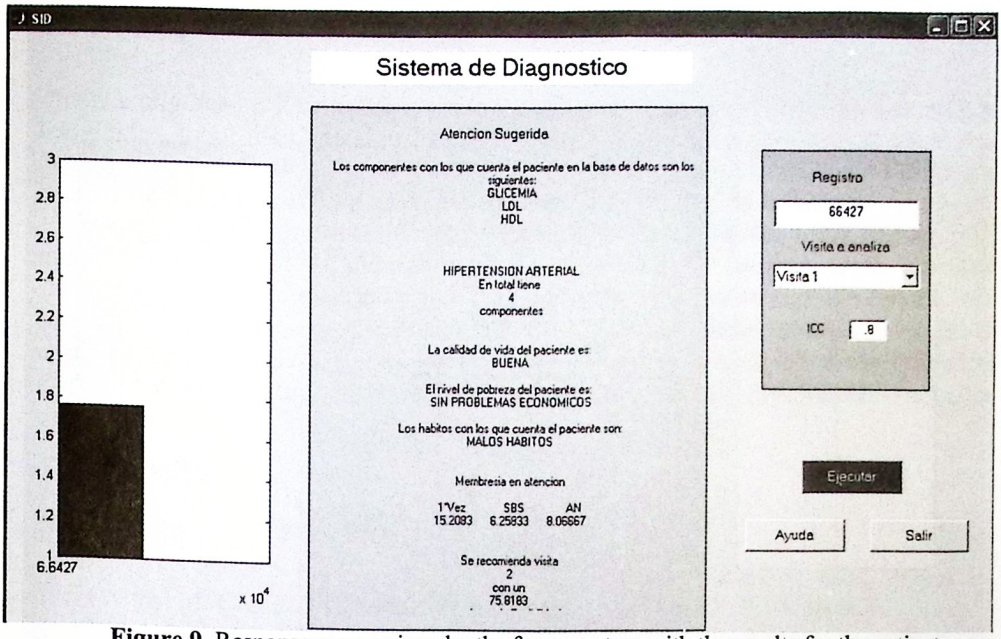


Figure 9. Response screen given by the fuzzy system with the results for the patient number 1. It should be noted the suggested therapy included at the bottom of the specific report.

The output in the graphic represents the defuzzification result, using the designed membership functions, the linguistic variables selected and the overall fuzzy inference model. The final value (1.7581) indicates the patient requires a new session to be attended with a medium urgency (possibility) determined by a 75.18 %. If the number in the response is 3, indicates the patient should be keep at the hospital to be observed by the medics and 1 with 0 % indicates the patient can be released for the treatment. To prove the workability of the suggested method, a second case is considered. The clinical values for this patient are:

- Sex: Female. She is a patient in death danger whit a lot of complications in her metabolic state. There is some evidence related with tumor activity, chronicle infection and a generalized organic fail. The given results are depicted in figures 10 (SM diagnose) and 11 (risk scale allocation).

The given results demonstrated this patient should be attended "before" the results derived before (patient 1), because the possibility or urgency to treat this patient is 90.91 %. At the same time, the results suggest a Risky Nutritional Support.

The analysis derived by the MS pre-diagnostiv system allows stating a well defined methodology to provide a range or degree of attendance and to define repeatability concepts to treat the MS illness, due the numerical algorithm introduced in this study.

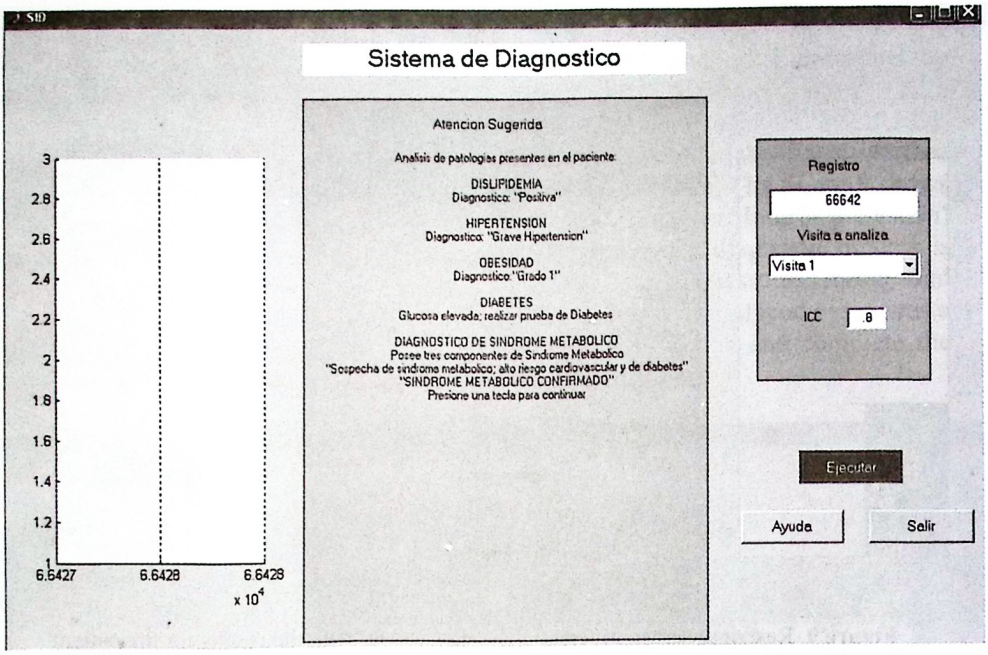


Figure 10. Results derived by the second patient's data from the diagnose algorithm.

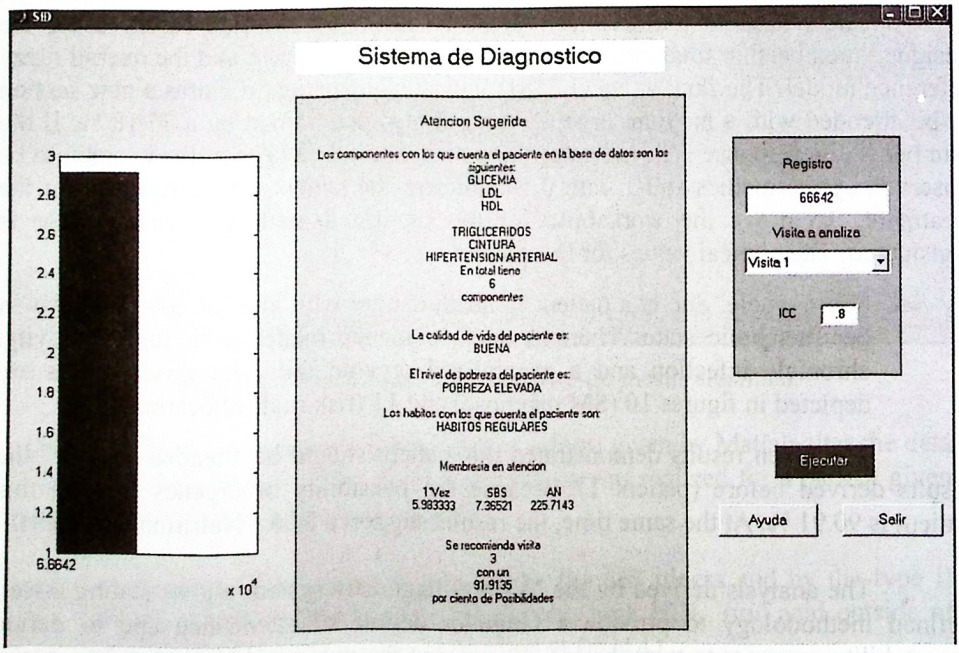


Figure 11. Results given by the risk analysis fuzzy system

4 Conclusions

The friendly design for the DB gives to the medic an easy way to acquire the enough knowledge about the clinical history, the lifestyle and some general aspect about the patient that the physician is attended. Besides, the complete elements considered in the DB fields allow defining a better manner to study the patient sufferings.

The linking process between the ODBC system interface (ACCES) and the numerical software (Matlab), give us the possibility to analyze (with advance statistics tools) and to treat the stored data using the well known decision capabilities of the hierarchical fuzzy logic method (a novel approach in the database field). This approach is really useful because many of the elements considered at the inference step are based on the medic opinions (they are with the INCMNSZ' nutrition department).

References

1. Castro, M. Gámez. "Historia Clínica", Reporte Clínico del HNCMNSZ
2. R. Porto. "Síndrome Metabólico. Enfoque Actual" , Revista Mexicana de Cardiología, 2001.
3. I. Chairez "Control inteligente aplicado a incubadoras neonatales", Tesis de Licienciatura, UPIBI, 2003.
4. C. Pérez, "Access 2003" , 2002, Mc. Graw Hill.
5. L. Zadeh, :A: " Fuzzy sets. " 1965 Inf. Cont. 8:338-353
6. H. Zimmermann. –J. " Description and optimization of fuzzy systems" Int. J. Gen. Syst. 2:209-215.
7. S. Gómez "Sistemas Difusos Jerárquicos"
8. L. Groop, M. Orho-Melander. The dysmetabolic Syndrome. J Intern Med 2001; 250
9. A. Zimmet "Definition, Diagnosis and Classification of Diabetes Mellitus and its Complications"
10. Isselbacher, Braunwald "Principios de medicina Interna", Mc Graw-Hill.

Rule-Based Expert System to Determine the Likely Diseases in an Air-Polluted Environment

R. V. Díaz[1, 3], Zenaida García[2, 3], Mateo Lezcano[2, 3] and Elena Rosa Dominguez[2, 3]

[1]rvdg@nuclear.inin.mx
[2]{zgarcia, mlezcano, erosa}@uclv.edu.cu

[1]Instituto Nacional de Investigaciones Nucleares (ININ),
Carretera México Toluca S/N, La marquesa Ocoyoacac, México. C.P. 52750
Tel.: +52 53 29 72 00 ext 2646, fax: +52 53 29 73 32

[2]Universidad Central "Marta Abreu de las Villas"
Carretera a Camajuaní km 5.5. Santa Clara, Villa Clara, CP 54830
Tel. 53 42 281194/ 281630, Fax: 53 42 281608

[3]Instituto Tecnológico de Toluca (ITT)
Av. Instituto Tecnológico s/n ExRancho La Virgen, Metepec, México CP: 52140
Tel. 01 722 208 7 224 ext 340

Abstract. In this paper we present an expert system which relies on production rules and aims to determine the set of diseases which might appear in a polluted environment. The main contribution of this study lies in the quality of the acquired knowledge by using rules as the inference engine. Experimental studies comprised the application of PIXE (X-ray Emission induced by Particles) techniques. The expert system is intended to be used by chemical analysts with an environmental profile, yielding an accurate representation of the knowledge base and leading to several important conclusions.

1 Introduction

Among the several branches that Artificial Intelligence encompasses, the Expert Systems (ES) have climbed to a unique position owing to their successful applications to a wide variety of problems as well the speediness in getting the problem solution and a higher quality of the solutions obtained [1, 2, 3]. Such features turn the ES into an indispensable tool for decision making in, for instance, the field of risk management, wherein quick and reliable decisions are needed. Thus the ES guarantees its inclusion within the set of strategies for risk management. A clear problem statement is fundamental when studying the atmospheric pollution bearing in mind that the main goal is to avoid the possible health risks by the contaminant powers present in it such as the suspended particles with an aerodynamic diameter less than or equal to 2.5 micrometers. ($PM_{2.5}$).

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 442–449

Received 16/06/07
Accepted 31/08/07
Final version 30/09/07

Considering that a great deal of countries, mostly located in Latin America, do require this type of research study, the envisioning and implementation of strategies tending to assure a right decision making is compelling, therefore achieving a better overall air quality and equipping the population with a fat better life style. In Mexico, relying on trustable data to decision making is of the utmost importance so as to enhance, for example, the development of Mexican standards of the air pollution as well as counting on sound foundations for the development and evolvement of programs such as the one of the Metropolitan Indicator of Air Pollution (IMECA, acronym in Spanish) that are suitably applicable without the need of copying models coming from other countries having a very different population in terms of customs and idiosyncrasy [5, 6]. The expert system created for such purposes allows determining the likely diseases which may arise due to the chemical elementary composition contained in the $PM_{2.5}$ and will facilitate the decision making process in such settings.

2 An Overview to the Environmental Problem Leading to the Expert System

The elementary chemical composition of the $PM_{2.5}$ is a potential danger for the population because of the inhalation risk. This was an essential requisite to be considered for building the expert system. This section focuses on describing the characteristics and methodology taken from the environmental domain.

2.1 Main Features

a) The sensitive population group was represented by children from 2 to 6 years old as well as those ranging from 6 – 12 and adults with an average age of 70 years old.
b) The sensitive groups endure, on a daily basis, contamination episodes mostly provoked by the $PM_{2.5}$ being present in their dwelling place [7, 8, 9].
c) Pollution sources: The exposition origin to these substances is of natural and anthropogenic genesis (lots of industries, automobiles, etc.)
d) Exposition Point: The atmospheric air.
e) Conveying Mechanism of Contaminant Powers: The air.
f) Exposition Ways: The inhalation of the polluted air.
g) Period of Observation: Dry, cold season.

2.2 Methodology

a) A place of interest where the contaminant $PM_{2.5}$ substances reside may damage the sensitive population groups.
b) Samples of the contaminant powers were picked up and subsequently analyzed by considering the Protons-Induced X-Ray Emission (PIXE) technique. The achieved

elementary chemical composition is: S, Cl, K, Ca, Ti, V, Cr, Mn, Fe, Ni, Cu, Zn, Br and Pb [8, 9].

c) The inflicted population risk by the inhalation of this contaminant was determined after regarding the Exposition Dose (ED) and the Reference Doses (RD) according to [10, 11]. The expression governing this step is the ensuing:

Risk (R) = (DE) (mg/kg/day) / DR (mg/kg/day)

The outcomes for the three sensitive groups are pictured in Table 1. From the obtained elemental risk, the likely diseases that may be caused by an element either alone or in conjunction with others hitting the sensitive groups are calculated. In total, 79 relationships were spotted. Some of them are illustrated in Table 2. After arranging the knowledge expert, the Expert System was built by using production rules as the Knowledge Representation Form (KRF).

	RISK (R) = DE / DR		
ELEMENT	ADULTS	6-12 year-old Children	2-6 year-old Children
	DS (mg/kg/day)		
SULFUR	0,03	0,09	0,09
CHLORINE	0,22	0,69	0,68
TITANIUM	0,16	0,16	0,16
VANADIUM	7.84E-04	2.45E-03	2.41E-03
CHROMIUM	0,02	0,05	0,05
MANGANESE	0,04	0,13	0,13
NICKEL	3,66E-05	1,14E-04	1,13E-04
COPPER	1.33E-04	1.33E-04	1.33E-04
ZINC	1,50E-05	4,70E-05	4,63E-05
LEAD	0,68	0.68	0.68
Overall Exposition Index	1,15	1,81	1,79

Table 1. Risk representation for the three sensitive groups.

3 Results and Discussions

The aforementioned methodology gave rise to the expert system which aims to establish a relationship between elemental chemical composition contained in the suspended particles $PM_{2.5}$ and the several diseases which those may bring about. The Rule-Based System (RBS) is made up of 480 rules from which 243 of them are used in the diagnosis of the possible diseases and the rest allow the functioning logic to perform well. The Inference Engine (IE) was supplied by the WUCShell system version 1.0 [15]. Figure 1 shows the functional structure of a Rule-Based Expert System using WUCShell.

An illustration of the rules section of the ES is presented as follows:

```
Rules
  Rule 7
    If element = 'S-Ni'
    Then
      SNi_Adult := 17.1*concentrationS+61*concentrationNi
  End;
  Rule 8
    If SNi_Adult > 1
    Then
      Disease: = 'asthma'
```

DISEASE	ELEMENTS [10, 12, 13, 14]
Asthma	S, Ni
Hormonal metabolism alterations	S
Bronchopneumonia	S
Cardiac damages	S, Cl, Ni
Damages on liver functions	S
Damages on kidney functions	S, Pb
Headaches.	S; Cl, Mn, Cu
Thoracic pain and obstruction.	S, Cl,
Bronchitis	S, V, Mn, Cl, Ni
Fever	S
Nose irritation	S, V, Cr
Throat irritation	S, Cl, V
Respiratory seizures	S, Ni
Death	S, Cl, Cr, Pb
Itchiness or burning sense of nose and throat	Cl

Table 2. Disease-elements relationships.

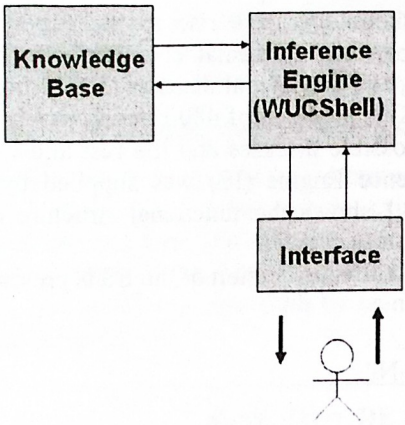


Figure 1. Functional structure of a Rule-Based Expert System using WUCShell.

Rule 7 expresses the calculation of the sum of the elemental concentrations for sulfur (S) and nickel (Ni). Rule 8 compares the sum of the risks with one (1). If the risk exceeds one due to the sum of S + Ni, then the chances are that the adults come down with asthma because of the inhalation of S and Ni. That possibility is null the other way around.

The ES attributes exhibited in the rules conclusions are inferables, i.e. they have to be checked across the inference engine. Other attributes are asked by the user and that's why the ES will prompt several times for the specification of the proper values as the inference process goes on. An excerpt of the latter type of attributes can be observed in the following lines.

```
Asks
  ASK element:
    'Por favor seleccione uno de los compuestos que se le muestra.'
    Domain
      'S', 'S-Ni', 'S-CL-Ni', 'S-CL', 'S-Pb', 'V', 'S-CL-Mn-Cu',
      'S-V-Mn-CL-Ni', 'S-Mn-Ni', 'S-V-Cr', 'S-CL-V', 'S-CL-Cr-Pb',
      'CL', 'CL-Ti-V', 'Ti', 'Cr', 'Cu', 'Mn', 'Ni', 'Cu-Zn', 'Pb'
  ASK concentrationS:
    'Especifique la concentración química del azufre (mg/m3):'
    Because
      'Con el uso de este sistema usted puede determinar el riesgo /n',
      'que enfrenta la población al respirar diferentes tipos de    /n',
      'contaminantes.                                               /n',
      'Debe especificar la concentración química del azufre para   /n',
      'el cálculo del índice de riesgo./n'
  ASK concentrationNi:
'Especifique la concentración química del níquel (mg/m3)'
```

The WUCShell inference engine is an objective-driven search tool by using the depth-first search method. The action block of the knowledge base is displayed next.

Actions
Find disease
End.

This is the starting point where the inference process begins. Notice that the sole objective to be inferred is the disease. A search process is launched by the inference engine which looks for that target alone, finishing only when a value is assigned to such attribute (successful test) or arriving to a state wherein no more rules are explorer (failure test). Some of the windows of the user interface of the WUCShell tool involved in the execution of the ES for a sample where the user picks up both the sulfur and nickel components (S-Ni) are portrayed in Figure 2.

When considering the multiple choices presented in Table 2 for the three sensitive population groups, it was inferred that none of the cases residing in the knowledge base are at stake of enduring any of such diseases. The only combination for which the likelihood of enduring any disease for population groups 2 and 3 (children-related) was that of (S, Cl, Cr, Pb). The disease in question is death. The same combination of elements is harmless for adults. Looking at the results bore by the ES when regarding a single element, it is expected that there is no chance of suffering any disease since the elementary risk is low. However the synergism of various elements triggers a risk increase which, in accordance with the literature, the risk is higher than the sum of the individual risks. The actual risk value owing to the synergic properties of the elements was not regarded in this study. The disease caused by the conjunction of elements S, Cl, Cr and Pb yield a risk having a sum greater than one and hence the aforementioned comment holds. With the aim of testing the ES for another study areas and considering elementary concentrations of the same contaminant (PM2.5), you can bear the results shown at Table 3.

DISEASE	ELEMENTS	LOCATION	LIKELIHOOD	REF.
Bronchitis	S, V, Mn, Cl	Ticomán (ZMCM)	NO	[16]
Bronchitis	S, V, Mn, Cl	Monterrey N.L. Mexico	SI	[17]
Kidney damage	S, Pb	Monterrey N.L. Mexico	SI	[17]

Table 3. Relation between the possibility of enduring diseases and several locations of the Mexican Republic (adults only)
ZMCM: Metropolitan Zone of Mexico City (acronym in Spanish). N. L. Nuevo León

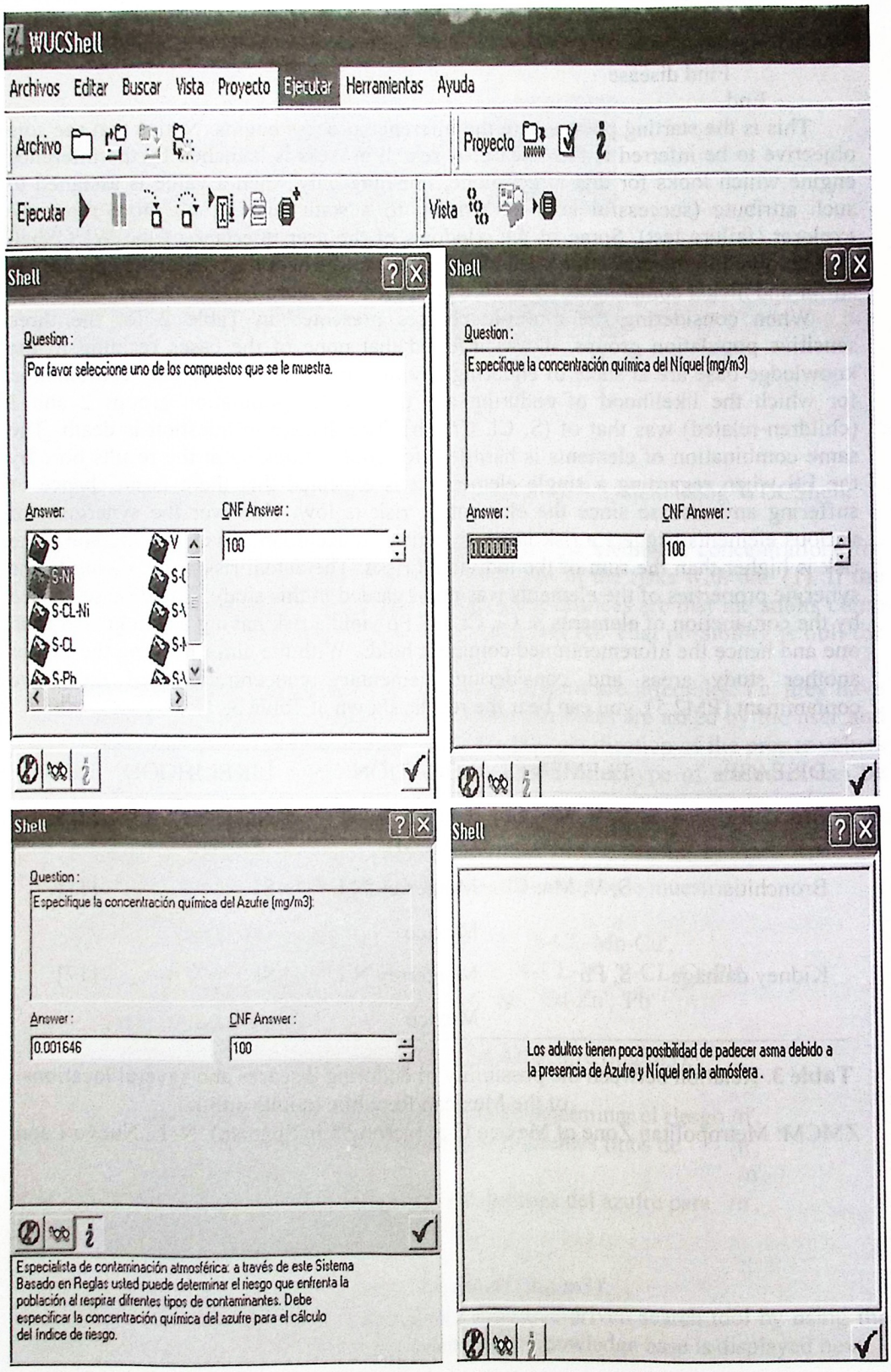


Figure 2. Different windows making up the execution of the ES for a sample.

4 Conclusions

1) The WUCShell tool allowed building the expert system in a fast and straightforward fashion.
2) Since the developed ES relies on sound data gathered by using reliable scientific methodologies, its own results are reliable as well.
3) Using this ES enables to bear quicker and more accessible results in such a way that it can be considered as an important tool for decision-making, i.e. an strategy to the risk management brought about by the inhalation of the $PM_{2.5}$.
4) The ES allows knowing the health risk (likely diseases the population may endure) as well as the emission sources coming from the chemical elementary composition. This was conducted in a way that allows exercising some degree of influence in the control or removal of the sources issuing this contaminant to the atmosphere.

References

1. Hatzilygeroudis I., (2004): An Expert System with Certainty Factors for Predicting Student Success. Lectures Notes on AI, Springer-Verlag.
2. Hayes, P., The Naïve Physics Manifesto. in: D. Michie (Ed.), Expert Systems in the Microelectronic Age. pp. 242-270, Edinburgh University Press, Edinburgh
3. Peña C., (2001): Toxicología Ambiental, Evaluación de Riesgos y Restauración Ambiental., The University of Arizona, 1996-2001
4. Mackenzie L., (2005): Ingeniería y Ciencias Ambientales, Mc. Graw-Hill Interamericana.
5. www.sma.gob.com
6. www.ine.com.mx
7. Cicero Sabido R., (1996): Contaminación Atmosférica y Salud en la Ciudad de México / Riesgos Ambientales Para la Salud en La Ciudad de México. Programa Universitario del Medio Ambiente, UNAM, p: 466.
8. Aldape F., (1991): Two Year Study of Element Composition of Atmospheric Aerosols in Mexico City. International Journal of Pixe, Vol.1, No.4 (1991)373-388.
9. Aldape F., (1993): Temporal Variations in Elemental Concentrations of Atmospheric Aerosols in Mexico City. NIM B75 304-307.
10. http://www.atsdr.cdc.gov/hac/PHA/isladeviquesspan/cdb toc ,html. ATSDR, Agency for Toxic Substances and Disease Registry, 2003. Evaluación de la vía del suelo / Campo de Bombardeo de la Isla de Vieques. Vieques Puerto, Febrero 7.
11. CEPIS/OPS, (2001): Curso de Auto instrucción de Evaluación del Riesgo Asociado a Contaminación del Aire.
12. http://www.lenntech.com.
13. EPA, U.S, (1991): Environmental Protection Agency. Risk Assessment for Toxic Air Pollutants: A Citizen's Guide - EPA 450/3-90-024, March.
14. www.toxnet.com. TOXNET, Toxicology Data Network.
15. Lezcano Brito, M. (1999): Manual para el uso de la máquina de inferencia WUCShell, Versión 1.0. Universidad Central de Las Villas, Santa Clara, Cuba.

Using Formal Concept Analysis to Leverage Ontology-Based Foot Acu-Point Massage Knowledge System

Kwoting Fang[1], Chingwei Chang[2] ,Yenping Chi[2]

[1]National Yunlin University of Science & Technology,Yunlin, 123 University Road, Section 3, Touliou, Yunlin, Taiwan
fangkt@yuntech.edu.tw
[2]National Chengchi University,NO.64,Sec.2,ZhiNan Rd.,Wenshan District,Taipei City 11605,Taiwan
{channing,ypchi}@mis.nccu.edu.tw

Abstract. The main purpose of this paper is to show the use of FCA to support the building of ontology-based Traditional Chinese Acu-Point with foot massage serving as an example. This knowledge-sharing platform may make expert's foot massage knowledge clearer, may connect with average people, and may provide patients with an opportunity to participate in and become knowledgeable of the whole prescription for his/her illness.

Keywords: Ontology, Formal Concept Analysis, Foot Massage, Information Retrieve.

1 Introduction

Given the striking progress of web service technology associated with the continuing rapid growth in knowledge management [7], it is imperative that we reduce the knowledge gap which exists between the patient and physician by using the medical knowledge sharing platform and by retrieving self symptoms, and as well as, through patients' themselves spontaneously learning medical knowledge. This may improve the interactive condition through the process of interrogation enquiry.

How to make information retrieval mechanism- friendly is always an important issue for system building and the creation of a User-Interface field. However, the way we use it to retrieve the information is still rife with problems [2]:

- User has to figure out the exact subject and keywords he/she would like to retrieve.
- The scope of retrieving is limited the same concepts but expressed by difference semantics are hard to find out with keyword searching.
- The accuracy of the retrieval results are difficult to establish.

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 450–459

Received 13/06/07
Accepted 31/08/07
Final version 22/09/07

How should information retrieval (IR) be set to more closely compare with the users' mindset in order to help the retriever more easily find the knowledge and to execute the function included in the system. The main objective of this study is to adopt formal concept analysis (FCA) to effectively break the restriction of IR, and to exploit the intention of the document by building a support structure and inter-document relationship automatically.

2.1 Using Ontology in Knowledge Retrieval

Ontology applied to knowledge retrieval definitely changes the strategy from a pull style to a push style [2]. Formerly the retriever inserted a certain keyword for looking up the information in the system, which is a kind of pull style of knowledge retrieval; at present, ontology concepts which provide the vocabulary for expressing personal interest profiles for information push service automatically deliver knowledge and information for the categories a user is interested in [1][8].

This study uses Formal Concept Analysis, FCA, as the mathematical principle to bring the benefits of FCA to a menu and folder hierarchy structure under an "user-unconscious" state. Because of the preceding, we can construct the menu settings more logically, and place the relative documents and knowledge so that all the information about foot acu-point massage is in the correct logical locations in the system.

2.2 Formal Concept Analysis

Formal concept analysis was originally developed as a field of applied mathematics based on the mathematization of concept and concept hierarchy, but after more than a decade of development, it has been applied in many quite different realms such as psychology, sociology, anthropology, medicine, biology, linguistics, computer sciences, math etc. From the philosophical point of view, a concept is a unit of thought, and is an idea for reinvigorating connections to our general culture by interpreting the theory as concretely as possible [6].

With the FCA advantage of the main ideas being quickly understood by formal lattices, linguists often characterize datasets by using distinct features, such as semantic components, phonemes or syntactical; or the terms of schema [9], linguistic applications often involve the identification and analysis of features, which were responded to by the participants, which explains the extraction schema from the interview of adopting new information technology more easily, therefore, the reason this study adopts formal concepts of FAC as an automatic technique to elicit the attributes of dependency schemas extracted from the documents and foot acu-point massage references.

3 Implementation the FACMKS

In this section, we will introduce the Foot Acu-Point Massage Knowledge System, FACMKS, in more detail, by following three principle steps. While put foot acu-point massage in practice of IR, usually involves the three steps listed below: (1)extracting a set of index terms that describe each collected document [5], (2)relating the document-terms generated from the step before with each other by concept lattice (3)visualization of the concept lattice constructed by step (2) to a structure diagram [4]. The outcome of each step may profoundly affect the overall effect of the application [2].

Such as, like how to organize queries about domain knowledge related articles or to understand the principles introducing system retrieval as constructed by the integrated software tool, Protégé. As we mentioned in the previous section, ontology could be seen as defining the set of concepts and the relationships between the concepts; such relationships facilitate knowledge-acquisition tools designed to be domain-specific, which allows domain experts to easily and naturally embed their knowledge of the zone. The resulting knowledge base can then be used as a problem-solving method to answer questions and to solve problems regarding the key point, according to the features of this study, Protégé [9]. This application is the end product which is created when the knowledge base is used in solving an end-user problem employing appropriate problem-solving, expert-system, or decision-support methods. Moreover, there are four strong points which can been achieved by Protégé,

- The modeling of an ontology of classes describing a particular subject
- The creation of a knowledge-acquisition tool for collecting knowledge
- The entering of specific instances of data and creation of a knowledge base, and
- The execution of applications. These characteristics being concluded are the reason we adopt Protégé to construct the framework of Search Data Management subsystem (SDM).

SDM is the core of the TCM knowledge retrieval platform for the building of foot acu-point massage ontology by Protégé for SDM; this design makes SDM have the capability for handing ambiguous information queries from the different users.

The functions of searching for specific news and for searching foot acu-point massage knowledge are providing users with a retrieving mechanism. By inserting the keyword that the user is interested in, the system will return the relative foot acu-point massage news report, article or documents.

3.1 Extracting a Set of Index Terms

There are many techniques that could improve the effect of a retrieval knowledgebase that we have discussed, but the knowledge itself always plays a major role in the knowledge system. It would be useless, if there were nothing in the knowledge sharing

platform, in the event of the latest technology being adopted. Emphasizing the source of knowledge itself is always the best policy for information retrieval system building.

With the Protégé application being developed for the creation of the ontology based system, in this step, we centre on creating the slots for each information item such as documents, clinical experience etc., with these slots also known as roles in description logics and relations in FCA construction and other object oriented notions. It also in some other formalism is called 'attributes'.

This study using foot acu-point massage related domain knowledge from literature, documents, CD etc., and creating foot massage, Foot massage ontology by the integrated software Protégé, the process of setup interfaces are as the lower diagram represents. For instance, Traditional Chinese foot acu-point argues that there are twenty four reflexology zones in human feet [5], Foot reflexology is "pressure therapy" and involves applying focused pressure to certain known reflex points located in the foot to cure or prevent disease, these section are adrenal gland, arm, bladder, brain, chest (lung), colon, diaphragm, ear, elbow, eyes, face, fallopian tubes, etc. as following figure shows.

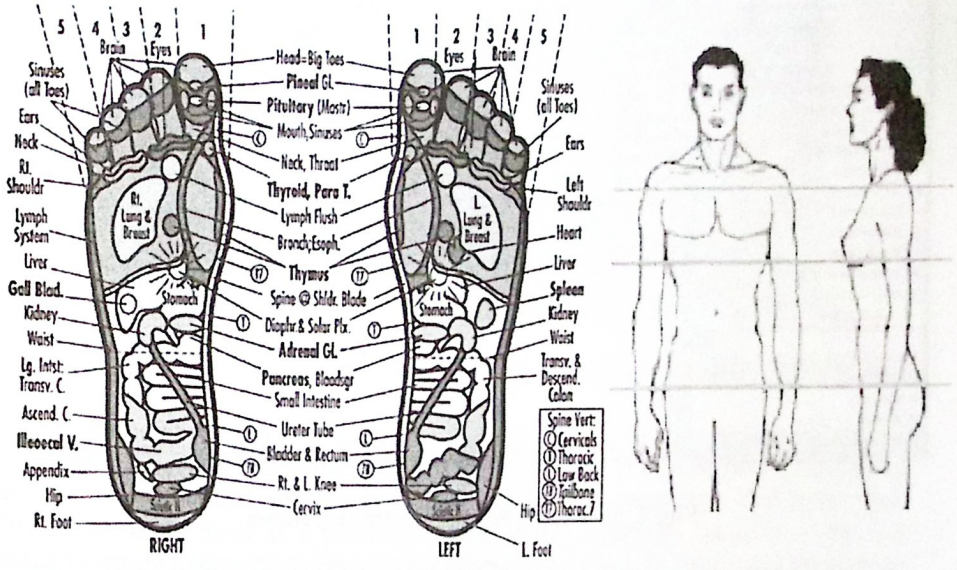


Fig. 1. Foot reflexologies mapping to different part of human body.

Foot reflexology is based on the premise that our nerve zones or reflex points go from the bottom of our feet to the top of our head, encompassing all vital organs on the way. A trained reflexologist can put pressure on different meridiens or energy lines on the sole and side of the feet to determine the cause of illness.

After identifying those reflexologies knowledge in foot massage domain, we set reflexology as kind of a super-class and then locate the detail region which can be treated as a sub-class of the super-class. Chinese foot massage says that foot reflexology is based on the premise that human nerve zones or reflex points go from the

bottom of our feet to the top of our head, encompassing all vital organs on the way, and focused pressure to certain known reflex points located in the foot can cure or prevent disease, therefore, when the study build up the foot acu-point massage ontology, we also set curative illness effect as a super-class, too. The Study adopted the integrated software tool -- Protégé to model of foot acu-point massage ontology of classes describing a particular subject, as following figure two shows is the interface of protégé software.

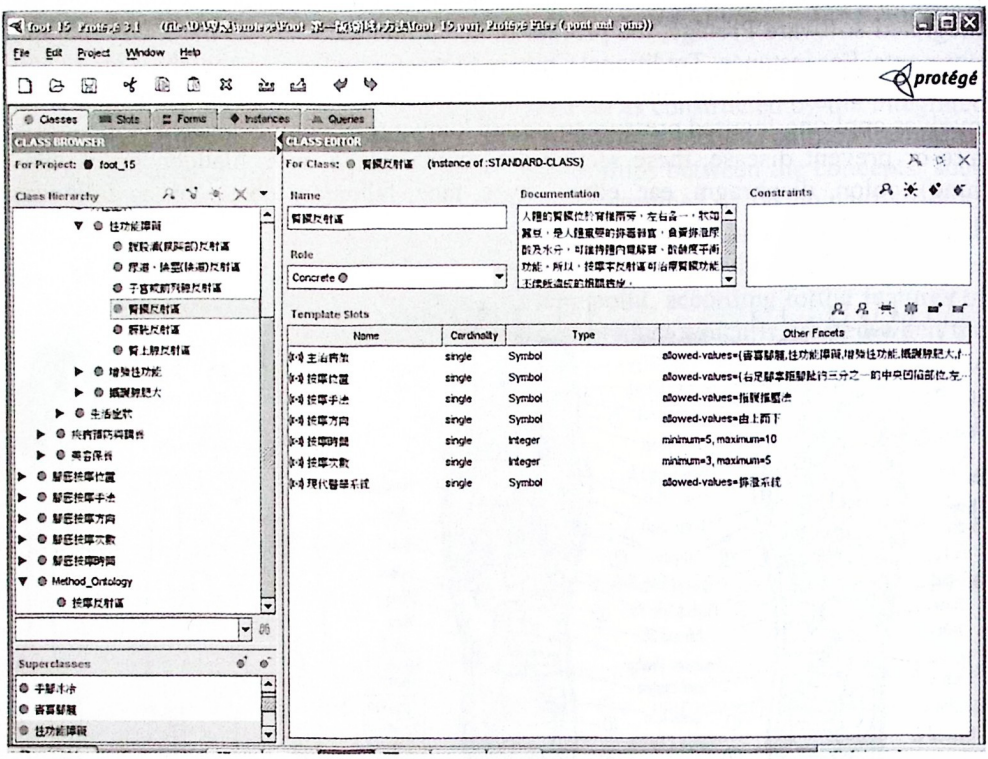


Fig. 2. Interface of Protégé constructing classes

As we reach the lowest level, where the scope narrows and becomes more specific, we find the class where the slots for foot acu-point massage are set. Slots are roles in description logics, and relations in FCA construction. Some FCA research label attributes as slots, which is the characteristic part of an object. For example, the kidney reflexology zone is described in terms of seven slots, which are massage techniques, curative illness effect, massage direction, position of reflexology zone, massage therapy, massage frequency and medical system. Different reflexologies may bring out a different nature and may be provided with different characteristics, such as massage different reflexology zones curing for different kinds of illness, or sometimes, massage certain reflexology zones could relax or reduce the painful of the illness, of course with the different massage frequency and techniques, these feature are be described in the

slot of the class. And massage techniques can also separated as stroking, rotation, pivoting, kneading and finger walking five sub-classes, the different part are displayed in Figure three and four.

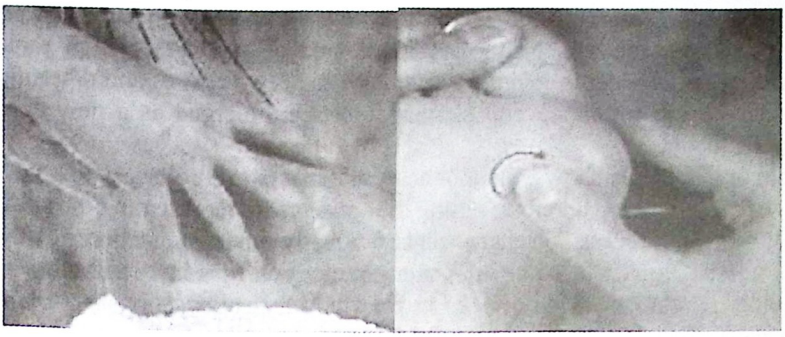


Fig. 3. Stroking technique and pivoting technique.

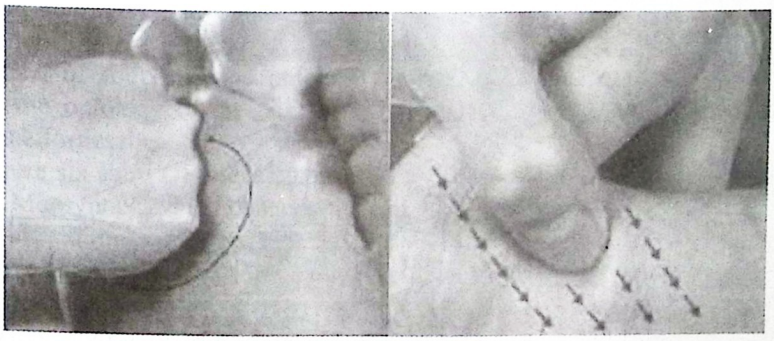


Fig. 4. Kneading technique and finger walking technique.

The slot curative illness effect, indicates results caused by massaging certain acu-points on foot and being able to treat all the different parts of the body through the feet, due to the fact that reflexology zones to all these different part exit in the feet, reflexology offers itself as a potential treatment for nearly all disorders. The slot massage therapy frequency represents how often and how much time should be spent to accomplish each round of massage therapy. The slot massage direction of massage specifies the direction for applying pressure. The slot massage technique, is the style of massage specifying the manner in which a massage therapist treats a patient, such as stroking, rotation, pivoting, kneading and finger walking which are mentioned before. Every foot reflex acu-point zone is a setup slot in Protégé for modeling ontology structure. Reflex acu-point zone is described by different slots in every aspect. Furthermore, building the relationship and constructing concept lattices must be based on these slots which belong to each reflex acu-point zone, as in the following steps.

3.2 Concept lattice Construction

Ontology describes the concepts in certain domains, and the relationship that exists between those concepts. In this section we will map the concepts into the appropriate classes. Classes are interpreted as sets that contain instances, with the construction of links between the concepts by the related degree of attributes which ranks each different concept, in order to find the common parts of two classes and determine their relationship. Furthermore, several similar objects are merged as a new concept in representing them.

As Figure five below shows, the foot acu-point massage ontology which we built is both large and complex. Therefore, setting the concepts for representing the sub-class is necessary. In this process, seven concepts gradually emerge and become the main essence of the foot acu-point massage ontology: "Massage technique class", "Reflexology class", "Massage direction class", "Medical system class", "Curative illness effect", "Massage frequency class", and "Massage therapy".

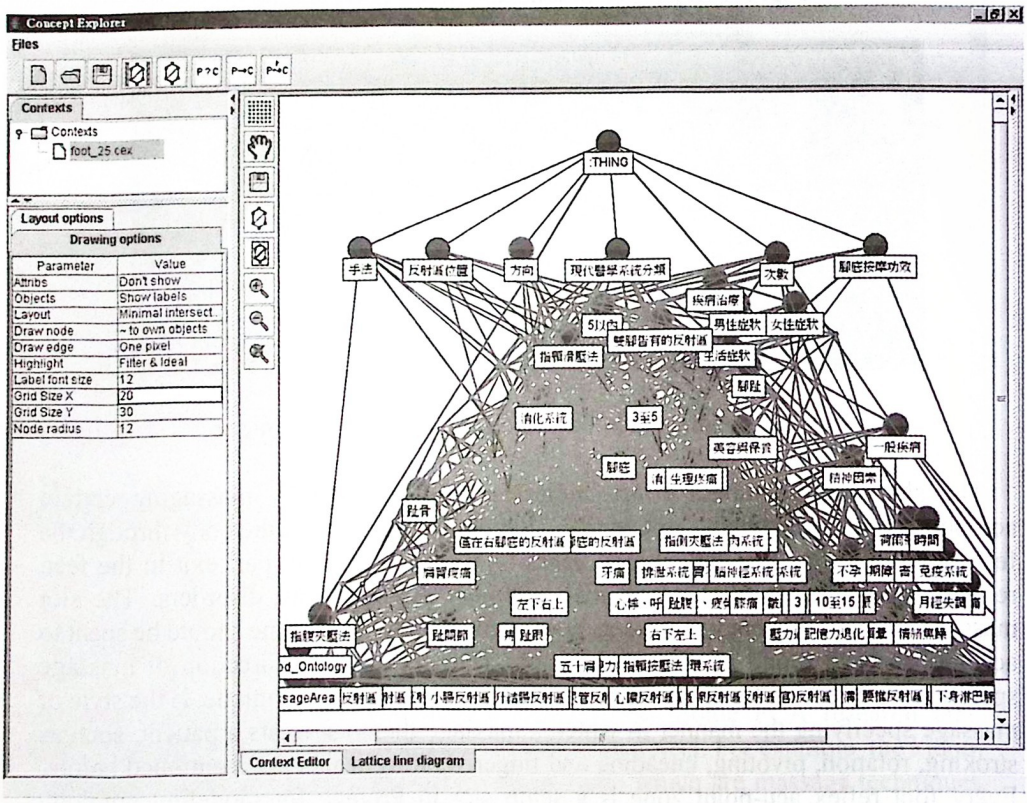


Fig. 5. Concept lattice of foot acu-point massage knowledge Ontology

Although the diagram above is difficult to read, its purpose is to show the result which was built via formal concept analysis. By setting the attributes and concepts through the use of mathematics, and making the facts more objective and reasonable, it

is not difficult to reassure people or for a beginner to understand, and let the people practice self-treatment, although reflexology is best given by a fully trained practitioner to obtain the maximum benefit from the treatment, it is also a method which can be used to a certain extent for self-treatment. Many people find the therapy from this foot acu-point massage knowledge sharing platform, which stimulates the reflexology zone by different massage techniques. Furthermore, it makes the constructed data-base cheaper and diminishes the need for experts.

3.3 Queries Outcome Visualization

By using the relationship between the concepts constructed in step2, users are allowed to view certain foot massage domain information in which they are most interested. Although this diagram may show only a partial Hasse diagram for foot acu-point massage domain, when selected by a user it will provide them with adequate general information concerning certain reflexology zone.

After constructing the ontology of certain domain data set up by the experts and builders, ontology is described by an accredited standard text format, the Extensible Markup Language, XML [3], which is generated by Protégé, while we finish the construction of foot massage ontology. It will simultaneously and automatically translate the ontology of the domain information to a webpage, answering users' questions effortlessly.

As figure six shows, Protégé compiled a domain information rule and relationship by the XML syntax, which contains the rule for the foot acu-point massage therapy method. Indeed, linking massage therapy, massage technique and curative effects with each other, and constructing different parts of the experts' clinical know-how into a knowledge- triangle, gives consideration to two or more aspects of health care, and the XML format is compiled automatically by the Protégé integrated software.

Protégé compiles the domain knowledge rule and relationship by XML syntax, which contains the rule of the foot massage therapy method, and includes the connecting curative illness, massage technique and curative effects with each other. This constructs difference aspects of experts clinical knowledge of how to be a knowledge- triangle, which gives consideration for two or more aspects of health care, and presents in the XML format which is automatically compiled by Protégé integrated software.

The outcome of the query displayed through the website return to users themselves may have advantages such as user-friendly feedback by multiple media, and the generation of dynamic results etc., However, FACMKS returns not only relative knowledge of requests, but also shows the whole picture of an FCA nested line diagram, and lets the retriever be aware of the knowledge he/she queried from that location of the whole foot acu-point massage knowledge ontology.

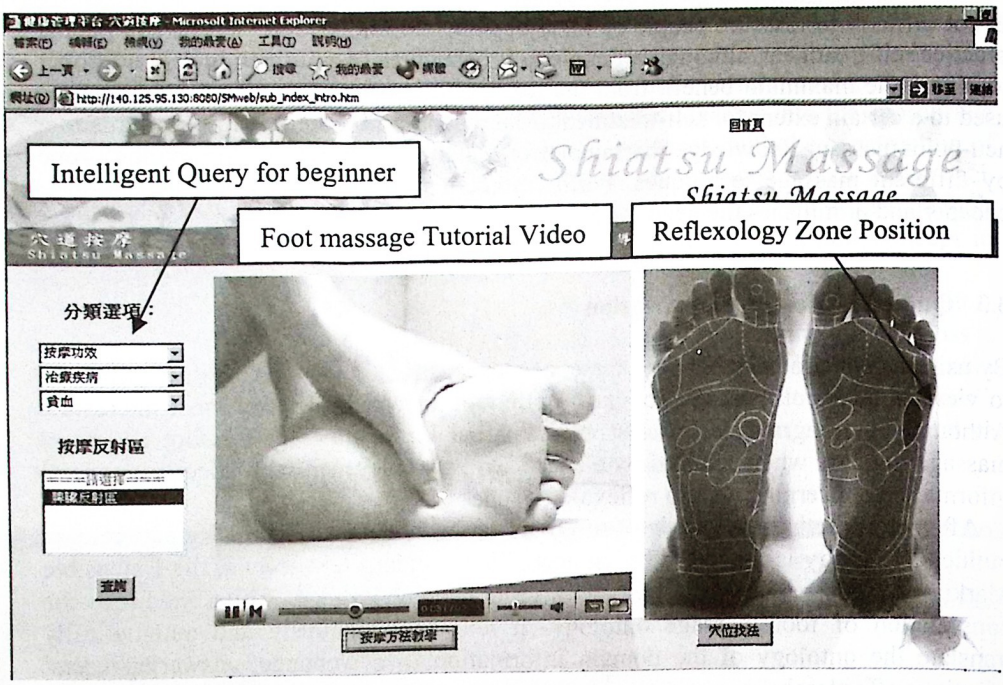


Fig. 6. Interface of "FACMKS" webpage

4 Conclusion

The goal of our developing FCA-constructed retrieve mechanism was to improve the traditional ways of managing information items (such as document, medical science literature) and domain knowledge used in the foot massage clinic, while minimizing the additional effort required from the information retrieve and content management. The above-mentioned also required us to put an emphasis on intelligent query interface—building the hierarchy with flexible information retrieve mechanisms through the ontology concept, which is quite different from before. It formulates user queries accordingly with ambiguities being avoided in the interpretation of both query and reply. Such normalization and retrieval promise precision in search beyond what is possible with the current keyword-based methods. Not having to be aware of the semantics of terms makes things much easier.

References

1. Andreas Abecker and Ludger van Elst, "Ontologies for knowledge management", Hand book of Ontology part 4, Springer Publishing, Inc.

2. Claudio Carpineto and Giovanni Romano, 2005, "Using Concepts Lattices for Text Retrieval and Mining", Formal Concept Analysis Foundations and Applications, Springer Publishing, Inc.
3. David Hunter, Andrew Watt etc., 2004, Beginning XML 3rd Edition, Wiley Publishing, Inc.
4. F. Vogt, C. Wachter, and R. Wille. TOSCANA – A graphical tool for analyzing and exploring data. In R. Tammassia and I. G. Tollis, editors, Graph Drawing' 94, pp 226-233. Spinger, Berlin, 1995.
5. Mingren Chen, The Principle of Foot Acu-Point Massage and Picture, 2004, San-Cai Publisher.
6. R.M. Botting,1998, "A formal and structured approach to the use of task analysis in accident modeling ", Academic Press,49,pp223-244
7. Rajiv Kishore, R.Ramesh, and Raj Sharman, 2003,"Computational Ontologies : foundations, representations, and methods ", Ninth Americas Conferences on Information Systems.
8. Tom R. Gruber, "Towards Principles for the design of Ontologies Used for Knowledge Sharing", Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.
9. Uta Priss, 2005, "Linguistic Applications of Formal Concept Analysis", Formal Concept Analysis Foundations and Applications, Springer Publishing, In.http://protege.stanford.edu/doc/users.html.

Descriptive Characteristics Generation and Selection for Acute Leukemia Subtype Classification from Bone Marrow Digital Images

M. Coral Galindo Domínguez[1], Jesús A. González[1] ,Leopoldo Altamirano Robles[1] and Ivan Olmos Pineda[2]

[1] Instituto Nacional de Astrofísica Óptica y Electrónica, Coordinación de Ciencias Computacionales
Calle Luis Enrique Erro No. 1 Sta. Ma. Tonanzintla Puebla México.
{cgalindo, jagonzalez, robles}@inaoep.mx
[2] Universidad Politécnica de Puebla
3er Carril del Ejido Serrano San Mateo Cuanala, Puebla, México.
ivanop_rkl@yahoo.com.mx

Abstract. In this paper we present a method for descriptive characteristics generation and selection applied to acute leukemia classification from bone marrow cells digital images. We obtain texture, geometric, statistical, and eigenvalues (PCA) as descriptive characteristics from our regions of interest (segmented cells). These characteristics are then used as input attributes to the data mining step (using different classifiers) to be able to classify acute leukemia families and then acute leukemia subtypes according to the FAB [1] classification. Our leukemia database also presented the class imbalance problem because of the proportion of cases of each subtype of leukemia, for which we applied over-sampling techniques. We made two types of evaluation, one with our domain expert and other using the cross validation technique with statistical significance. Results for each leukemia subtype were superior to 85% of accuracy.

Keywords: image processing, machine learning, acute leukemia.

1 Introduction

According to statistics shown by INEGI in 2005 [2], different types of cancer are becoming a large public health threat in Mexico. This is why physicians are working to improve its opportune detection, accurate diagnosis, and effective treatment. Leukemia is one of these public health enemies with a hospitalization rate of 49.7% for men and 18.4% for women and with a mortality rate of 6.3% for men and 5.5% for women.

Leukemia is a blood illness that produces an uncontrolled growth of white globules (a hematological cancer). Leukemia cells live longer than normal cells; they are different and also behave in a different way because they are unable to realize their main function (they should be antibodies to defend the organism). Leukemia is

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 460–470

Received 30/06/07
Accepted 31/08/07
Final version 11/09/07

classified as myeloblastic or lymphoblastic, depending on the type of white globules affected and can be presented in acute or chronic ways. Acute leukemia grows faster while chronic leukemia grows gradually. If leukemia is detected in its first stages, its treatment might be more effective. This is where we focus our research, in early leukemia detection through morphological characteristics. Although other detection methods exist, morphological studies continue being important because they can be done at low cost and without expensive specialized equipment. That is, in spite of the recent advances in hematological skills such as flux cytometry (with immunophenotype), and DNA analysis; morphological analysis of bone marrow smears (even peripheral blood) continues being the starting point to detect patients that suffer of blood disorders. A morphological blood study consists of a cells study where cells are counted to verify if its amount is abnormal (according to specified limits) and then the cells morphology is examined to detect abnormal cells, this is done by a chemist or hematologist with experience in the detection of the disease (the morphological analysis is difficult and this is why it requires of experienced people to perform it). The problem with this type of analysis is that it is slow and it is not as precise as required because of its subjectivity (there are differences among physician's diagnosis or a physician's accuracy to detect the disease may vary depending on his state of tiredness). On the other hand, an automated morphological analysis to detect leukemia only requires images taken from bone marrow smears which are analyzed without the subjectivity related to tiredness or experience and could also reduce costs and improve accuracy; it can even be used remotely for places of low resources and could also be used to train chemists and hematologists. A tool like this is necessary for the diagnosis and treatment prescription of many diseases such as leukemia.

The aim of this work consists on a process to generate a set of geometric, texture, and statistical descriptive characteristics of acute leukemia subtypes as well as eigenvalues (obtained from principal component analysis or PCA's) from the segmentation (using color and boundaries detection) of regions of interest (ROI) or white globules cells. We use the obtained ROI's characteristics to feed classification algorithms such as neural networks, decision trees, and support vector machines among others to create discriminating patterns to classify leukemia subtypes.

This paper is structured as follows; we first describe our database in section 2. We then show our methodology to classify acute leukemia subtypes in section 3. In section 4 we present our experimental results and in section 5 we show our conclusions and future work.

2 Acute Leukemia Subtypes Classification Methodology

The classification of acute leukemia subtypes is a difficult task because affected cells are very similar. For this problem, we present the methodology shown in figure 1 to be able to classify acute leukemia subtypes. According to our methodology (see figure 1), we first select the images of patients for which an acute leukemia subtype has been detected. We then start the pre-processing phase, which consists of the segmentation of ROI's using first a color segmentation algorithm (to eliminate the background) and

then with a boundaries segmentation algorithm to get the ROI's. We then eliminate red globules to only keep white globules, from which the domain expert identifies lymphoblasts or myeloblasts that we will use for training with our learning algorithms. We then extract descriptive features from each ROI, which will be used to find patterns in the following step. Finally, the classification module is the one that we use to classify acute leukemia subtypes.

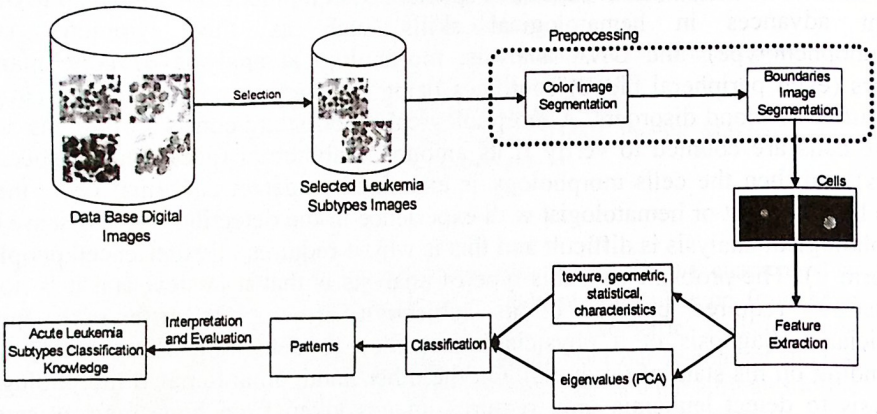


Fig. 1. Acute Leukemia Subtypes Classification Methodology

Our images database was created with the help of hematologists to select and classify acute leukemia cells in subtypes. In section 2.1 we describe our acute leukemia database.

2.1 Leukemia Data Base Description

Our database contains 1028 images from 74 patients. We have 415 images classified as Acute Lymphoblastic Leukemia (ALL) and 613 as Acute Myeloblastic Leukemia (AML) that were selected by domain experts. These images have a resolution of 800 x 600 pixels with a depth of 24 bits. We have only 11 cases of patients with ALL-L1, 8 patients with ALL-L2, 3 patients with AML-M2, 3 patients with AML-M3, and 1 patient with AML-M5. The number of cases presented before is related to the incidence of the disease subtypes. The number of images obtained of each leukemia subtype from the cases mentioned before is shown in table 1 as well as the number of ROI's (lymphoblast and myeloblast), which were labeled by the domain expert.

Table 1. Regions of Interest for every Subtype of Leukemia.

Acute Leukemia	Subtypes	Images	Regions of Interest
Lymphoblastic	L1	103	65
	L2	73	30
Myeloblastic	M2	57	38
	M3	48	26
	M5	11	10

2.2 Bone Marrow Smears Images Preprocessing

One of the most important phases of our methodology is the pre-processing of the images from which we obtain our ROI's through their segmentation so that we can obtain descriptive characteristics for each of them. Our pre-processing phase consists of four steps as shown in figure 2.

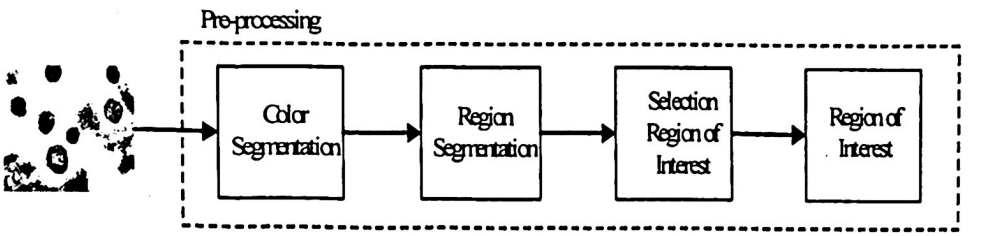


Fig. 2. Bone Marrow Images Pre-processing.

The image segmentation process consists of the division of an image in different ROI's, where each ROI has characteristics that distinguish it from others and each region is formed by a set of pixels. In this work we use a segmentation technique based on color and another to detect boundaries to separate leukemia cells (white globules) from the rest of the image. This is a very difficult task because in many cases the ROI's are overlaped and some others the boundaries of the cells are scarcely visible; furthermore, the colors of different types of ROI's (i.e. white and red globules) are very similar.

Color is an important descriptor in images analysis, it frequently helps to identify and extract regions of interest [3]. There exist different types of color formats such as RGB, CMYK, and Lab among others [4].

The RGB format works with three data channels where each them takes values from 0 to 255. An image in RGB is then composed of three bands with the primary colors of light Red (R), Green (V) and Blue (B) constructed with 8 bits per channel [4][5].

The Lab format consists of three channels, a channel L for Luminosity and two chromatic channels. The first chromatic channel, named A, ranges between green and red while the second, B ranges between blue and yellow. The luminosity component L ranges from 0 (black) to 100 (white). The chromatic components A and B range from +120 to -120 [4][5].

The original acute leukemia digital images are in RGB format, nevertheless; in order to improve the preprocessing efficiency we transformed them into the Lab format. That is, the RGB format is composed of three matrices, one for each channel, while in the Lab format we only work with a single matrix containing the information of channels A and B without considering the brightness channel (stored in a third matrix). Then, working with a single matrix reduces the work load. In figure 3 we show an image in the RGB and Lab formats. As we can see in figure 3, some ROI's (white globules) are very close to red globules (some of them even overlap), we can also see that the boundaries that delimit the cells are not very well defined and these are some of the problems that make the segmentation process difficult.



a b

Fig.3. Acute Lymphocytic Leukemia Subtype L1 Image. a) RGB Format. b) Lab Format.

2.1.1 Color Segmentation

As we mentioned before, for our color segmentation process we worked with the Lab format, omitting the luminosity channel. Figure 4 shows our general process for color segmentation, for which we used the K-means algorithm to group different colors of pixels in clusters to be used to find ROI's

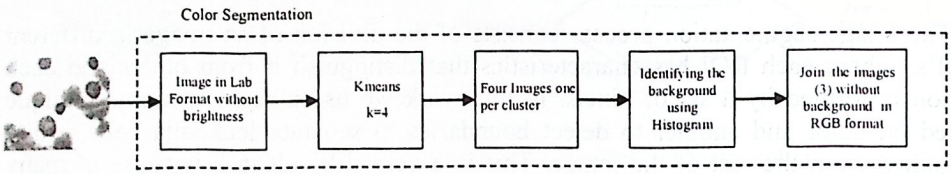


Fig. 4. Color Segmentation Process.

The K- means algorithm finds k groups, where each object belongs to one of the k groups. Each of these k groups is represented by a vector created as the average of the objects that belong to it [6].

In used the K-means algorithm [6] with the Euclidean distance and k = 4 to generate color groups. The chose k = 4 because the digital cell images of acute leukemia have 4 main tones, one for white globules, another for cytoplasm, one more for the background and finally one for red globules. We generated four images, one for each group as it is shown in Figures 5 a, b, c, and d respectively.



a b c d

Fig. 5. Acute Lymphocytic Leukemia Subtype L1 separated by color, with k = 4. a) White Globules. b) Cytoplasm. c) Background. d) Red Globules and Platelets.

For each of the images generated, we obtain the histogram, this will allow us to determine which images contain the ROI's (white globules), and which of them contains the background. In the last step we join three of the resultant images to create

an image without background in RGB format (we transform our Lab images to RGB format before mixing them) to get an image as shown in Figure 6.

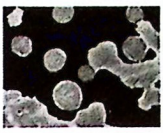


Fig. 6. Acute Lymphocytic Leukemia Subtype L1 separated by color, without background.

2.1.2 Regions Segmentation

After we eliminated the image background through color segmentation we perform a ROI's segmentation. In figure 7 we show our ROI's segmentation process.

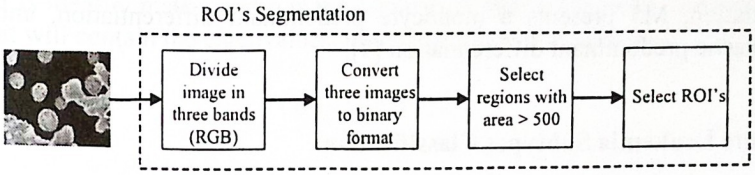


Fig.7. ROI's Segmentation Process.

Since we need to obtain descriptive characteristics from our ROI's, we use the images in their original format (RGB) and we separate them in their three channels to obtain their eigenvalues for each channel in gray format.

In this process we take the image segmented by color (without the background) and separate it in its three bands (RGB). Each image (3) is then converted into binary format; we then find the boundaries of each region and assign an index to each of them as shown in Figure 8 where the different tonalities indicate the different identified regions, we only keep regions with an area larger than 500 pixels because smaller areas correspond to noise. Finally, with the help of the domain expert we decided which of the identified regions corresponded to our ROI's (white globules) were.



Fig. 8. ROI's Segmentation of an Acute Lymphocytic Leukemia Subtype L1 Image identified by color.

ROI's in our case correspond to white globules from which we can identify acute leukemia (myeloblastic or lymphoblastic). The previous process is repeated for each

image in our database or for each new image to be classified. Once we have segmented our ROI's, we need to obtain their descriptive characteristics to be used as input to data mining algorithms and find patterns that distinguish among the different subtypes of leukemia.

For our classification process we use the French-American-British classification (FAB) based on morphologic and cytochemical characteristics that considers the maturity level of the cells (white globules) and the cellular line to which the white globule belongs (i.e. blasts, erythroblasts among others) [1]. According to the FAB classification [7], acute lymphocytic leukemia and acute myelogenous leukemia are divided in three and six groups respectively.

Acute lymphocytic leukemia splits into three types: L1, L2 and L3 according with the occurrence of cytological features and to the degree of heterogeneity in population of leukemia cells. Acute myelogenous leukemia splits into six principal types according with the direction of the differentiation along one or more cellular lines and their maturity degree. AML subtypes M1, M2, and M3 can exhibit a differentiation with granulocytic predominance. M4 presents a granulocytic and monocytic differentiation, M5 presents a monocyte predominant differentiation, and M6 an erythroblastic predominant differentiation [7].

2.3 Acute Leukemia Subtypes Classification

In order to recognize the subtype of each ROI (white globules in our case), we obtain descriptive characteristics that help to discriminate among them. In this work we use three types of characteristics: texture, geometric and statistical (we also obtain eigenvalues from PCA's but we do not consider them as descriptive in the same sense of these). A characteristic of a ROI is a primitive distinctive attribute. Some characteristics are defined by the visual appearance of an image, whereas others result from some specific manipulation.

Our process to obtain descriptive characteristics from our ROI's begins with the conversion of the original region of interest to tones of gray, from which we obtained characteristics (see Table 2) such as Perimeter, Area, Major and Minor Axes, Gray Threshold, Solidity, Total Number of Pixels, and Average among others [12]. Some of these characteristics were obtained using procedures of the Matlab [13] for images library, while others were programmed (also in Matlab). As we mentioned before, we obtained texture characteristics, which are defined according to the interrelation between arrangements of pixels in the image. These are seen in the image as changes in intensity or in tones of gray. We also generated statistical characteristics from the image pixels distribution. Finally we obtained geometric information from the ROI's shape.

Table 2. Some Descriptive Characteristics Obtained from ROI's

Texture Characteristics	Geometric Characteristics	Statistical Characteristics
Threshold of Gray, Maximum value of Gray, Minimal value of Gray ...,	Perimeter, Area, Center, ...	Average, Standard Deviation, Variance, ...

We obtained 30 descriptive characteristics for each region of interest. We now need to find those characteristics that help us to discriminate among the different acute leukemia subtypes in order to achieve good classification accuracy [11]. Looking for other types of characteristics that help us to improve our classification accuracy, we explored the use of principal components analysis to find the eigenvalues to describe our ROI's as described in section 3.2.

3.2 PCA

The aim of Principal Components Analysis is to reduce the dimension of a set of p variables to a set of m of minor size.

The set of Eigen values took as attributes of entry for the classification. The process that we follow to obtain the eigenvalues is described in [8] in contrast that we use the roots of the typical equation (eigenvalues) as attributes. To obtain the Eigen values and eigenvectors solve the characteristic equation [8][9]showed in the equation 2, where I is identity matrix and ρ is the correlation matrix (3) and λ is the questions matrix that will contain the eigenvalues on having solved the equation.

$$| \rho - \lambda I | = 0 \tag{2}$$

$$\rho = \frac{\mathrm{Cov}(x)_{i,j}}{\sqrt{\mathrm{Var}(x)_{i,i}}\ \sqrt{\mathrm{Var}(x)_{j,j}}} \tag{3}$$

We calculate three matrix of correlation, for each image of three channels (RGB), and we obtained eigenvalues and eigenvectors resolving the characteristic equation (2). As descriptive attributes or characteristics we use the eigenvalues, only we select eigenvalues that accumulate the 80% of changeability, in our case were 10 eigenvalues for each band (30 eigenvalues for each ROI) to comply with this condition [10]. Once we obtained Eigen values we did experiments with the different characteristics obtained of ROI's (geometric, texture, statistical, and eigenvalues) to obtain the best possible accuracy to classify the different sub–types of acute leukemia myeloblastic or lymphoblastic.

3 Experimental Results

In this section we describe our classification experiments using the descriptive characteristics obtained from the bone marrow smears using different machine learning algorithms implemented in the WEKA [14] data mining tool. We used a binary classes technique for It is important to mention that our databases presents the class imbalance problem (i.e. we have 38 examples of myeloblastic leukemia subtype M2 and only 10 examples of myeloblastic leukemia subtype M5) between leukemia subtypes and then we included oversampling techniques only for texture, geometric,

and statistical characteristics for the L2 (100% of oversampling to get a total of 60 examples), M3 (100% to get 52 examples), and M5 (300% to get 40 examples).

In our oversampling proces each subtype (only for the L2, M3, and M5 subtypes) we divide the original set in 10 groups and we keep one of those subgroups to validate (or test) the model obtained with the rest of the groups (9 groups), we applied the oversamplig process only for the training phase (with SMOTE we generate new examples).

We performed two classification experiments for the classification of acute leukemia subtypes. In the first experiment we only considered texture, geometric, and statistical characteristics (30 attributes). In the second experiment we only used PCA eigenvalues (also 30 attributes). For our results evaluation we performed a 10-fold cross validation.

3.1 Geometric, Texture, and Statistical Based Classification

For our first experiment, the task consists on classifying acute leukemia subtypes, we have 5 classes; two of them correspond to the lymphoblastic acute leukemia family (L1 and L2) and three to the myeloblastic acute leukemia family (M2, M3, and M5). It is important to mention that in order to do the acute leukemia subtypes classification, we first need to perform a family classification (between lymphoblastic and myeloblastic) and then we do a classification among the leukemia subtypes from each family. Then we created two data groups, the first one containing the L1 and L2 subtypes and the second one with the M2, M3, and M5 subtypes. Finally, for the M1, M2, and M3 subtypes classification we used a two class scheme (we created three data subgroups with a positive and a negative class where the positive class was M2, M3, or M5 respectively for each group). This was the scheme that gave us the best results for the AML subtypes.

The best results were obtained with the IBk, Random Forest Ensemble, AdaBostM IBk ensemble, and the AdaBoostM Bayes Net TAN ensemble algorithms as shown in table 3. For the IBk algorithm we always set k to 3 (we tested with k from 1 to 6 but k = 3 led us to the best results). For all the experiments we ordered the 30 attributes according to their correlation with the class, we did this using a ranker filter of Weka. For the IBk experiment for lymphoblastic leukemia we additionally performed an attributes selection considering attributes: perimeter, distance, area, bounding box, mayor Axis, minor Axis, convex Area, diameter of a circle, entropy, gray threshold, total histogram, max value of histogram, max value of histogram, average, standard deviation, variance. As we can see in table 3, we obtained better accuracy for the myeloblastic leukemia subtypes and this can be a consequence of the high similarity between the L1 and L2 subtypes. We can also notice the use of ensembles improves the classification accuracy. We also found that using oversampling techniques improve the classification accuracy for the M5 leukemia subtype up to 95.38%. The classification accuracy average by family was of 85.2% for acute lymphoblastic leukemia (L1 and L2) and 91.7% for acute myeloblastic leukemia (M2, M3, and M5).

Table 3. Classification Results using Geometric, Texture, and Statistical Characteristics

Acute Leukemia Subtypes	Classifier	Accuracy
L1, L2	RandomCommittee.RandomForest	86.40
	IBk	84.00
M2 and the others M's	AdaBoostM.IBk	91.53
M3 and the others M's	IBk	88.46
M5 and the others M's	AdaBoostM.BayesNet.TAN	95.38

3.2 PCA's Classification

For these experiments we used eigenvalues as attributes, which are already ordered according to the variability that they represent, then it was not necessary to apply an attribute selection algorithm; we only used 10 eigenvalues for each band of the RGB format.

Table 4. Eigenvalues Results Classification

Acute Leukemia Subtypes	Classifier	Accuracy
L1, L2	trees.ADTree	87.20
M2 and the others M's	AdaBoost.IB1	90.76
M3 and the others M's	AdaBoost.KStar	90.00
M5 and the others M's	AdaBoost.SMO	98.46

In this case, the best results were obtained with the Adtree, AdaBoost.IB1, AdaBoost.Ktar and AdaBoost.SMO and Decision Table algorithms. We tried other algorithms but we only show the best results in table 4. As we can see we obtained better accuracy results for the acute myeloblastic leukemia subtypes (as in the previous test) and this suggest that it was not possible to completely differentiate the lymphoblastic leukemia subtypes neither with the geometric-texture-statistical characteristics nor with eigenvalues because of the high similarity between them. The classification accuracy average by family was of 87.2% for acute lymphoblastic leukemia (L1 and L2) and 93.40% for acute myeloblastic leukemia (M2, M3, and M5).

4 Conclusions and Future Work

In this paper we presented a methodology to classify acute leukemia subtypes using different types of characteristics: geometric, texture, statistical, and eigenvalues. This method is based on a morphological and statistical analysis of white globules from bone marrow smears. The implementation of this process can be used as a diagnosis support tool for the identification of acute leukemia subtypes and to provide more accurate treatments to patients.

We would also like to compare our system with others but we have not found any to do it yet. We will continue looking for other systems to do it. What we will do is to compare our results with data from our domain experts. What we are now comparing

is how the use of two different types of characteristics performs in the process of acute leukemia subtypes classification.

We could see that both descriptive characteristics (geometric, texture, and statistical) and eigenvalues (from PCA) produce high accuracy results around 85% for the L1 and L2 subtypes and around 91% for the M2, M3, and M5 subtypes.

The accuracy achieved up to now is enough for a diagnosis support tool but we think it can be improved through the use of new characteristics, enhancing the use of oversampling techniques, and the combination of both types of attributes (geometric, texture, and statistical combined with eigenvalues).

Acknowledgments

I wish to thank Dr. José E. Alonso and Chávez, Dr. Rubén Lobato Tolama, Chemist Laura O. Olvera Oropeza for helping us with the classification of the images of the leukemia database and the Mexican Institute Social Security (IMSS) San Jose in Puebla for providing the data and Blanca Aurora Morales Gonzalez for contributing to the creation of the database. Finally, I especially thank CONACYT for my scholarship Number 202001.

References

1. Bennett J. M. et al. Proposals for the classification of the acute leukemias. French-American-Brithis (FAB) co-operative group", Br J Haematol.; Vol 4, No. 33(1976), 451-458
2. http://www.inegi.gob.mx/inegi/default.aspx
3. Pajares G., M. de la Cruz J., Vision by Computer, AlfaOmega 1st ed, 123
4. Gonzalez R., Woods R., Eddins S., "Digital Image Processing Using MATLAB", Pearson-Prentice Hall,1st ed, 94, 194, 195, 206, 463, 478, 485
5. http://www.desarrolloweb.com/articulos/s1778.php
6. J. B. MacQueen (1967): Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297
7. McDonald G, Paul J. Cruickshank B., "Hematology Atlas, Pan-American", 5th ed, 85
8. Jackson J., "A User's Guide to Principal Components", Wiley-Interscience, 1st ed, 7, 64
9. Jollife I.T., "Principal Component Analysis", Springer, 2nd ed 1, 5, 21
10. Sanei S., Lee T., Cell Recognition Based on PCA and Bayesian Classification, 4th International Symposium on Independent Component Analysis and Blind Signals Separation(ICA2003), Nara Japan (2003)
11. Scotti F., Automatic Morphological Analysis for Acute Leukemia Identification in Peripheral Blood Microscope Images, IEEE International Conference on Computational Intelligence for Measurement Systems And Applications, Italy(2005)
12. Osowski S., Markiewicz T., Marianska B., Moszczyński L., Feature Generation for the Cell Image Recognition of Myelogenous Leukemia, 12th European Signal Processing Conference (EUSIPCO). Vienna(2004)
13. Matlab user manual – Image processing toolbox, MathWorks, Natick, 1999
14. http://www.cs.waikato.ac.nz/ml/weka/

An Electronic Game with Speech Recognition for Children with Problems of Speech

Carlos Miranda-Palma, Reyna Camal-Uc, José Cen-Magaña, Cinhtia Gonzalez-Segura, Sergio Gonzalez-Segura, Manuel Escalante and Jorge Ríos

Universidad Autónoma de Yucatán, Facultad de Matemáticas, Unidad Tizimin, Calle 48 A s/n x 31, 97700, Tizimín, Yucatán, México
cmiranda@uady.mx, {camaluc, cmagana}@tizimin.uady.mx, {gsegura, sgsegura, etorres, rmatine}@uady.mx.

Abstract. In this work, we describe an electronic game (Memory Game) with speech recognition for stimulation on children at scholar age with problems of speech. We present the characteristics and elements of electronic game, as well as its integration with speech recognition system. For developed of the electronic game we used the Language C and the SDL graphic libraries and for the acoustic models we used the HTK software of speech recognition.

Keywords: Electronic game, speech recognition, children, problems of speech.

1 Introduction

Speaking is a tool that human being uses to exchange information. However, there are people with speaking disorders, and it is difficult or impossible to them to communicate in this way.

A speech and language disorder refers to the problems of communication or other areas related to it, for instance the motor oral functions. These disorders vary from simple substitutions of sound to the disability to understand or use language to speak or feed [1].

The communication disorders have the potential of isolate to the individuals in their social and educational environment, and because of that it is necessary to find out mechanisms to increase life-quality of people with special needs, so they can contribute to society.

Technology can help to those children whose physical conditions make the communication difficult. The use of electronic systems of communication let people who don't talk or have severe disabilities in speech increase their participation [1].

Because of the importance of speaking in people's life, the development of systems able to accept a voice signal as input to do a determinated task has become a big challenge. Actually, great quantities of systems which use human voice are being developed, the ITESM Campus Cuernavaca that by means of speech commands a mobile robot is manipulated [2][3]. At IIMAS, from UNAM, had been developed several projects that included speech communication as a interaction form [4][5].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 471–479

Received 18/06/07
Accepted 31/08/07
Final version 30/09/07

In the same way, in the latest years had been incremented the development of applications for people with some kind of disability, For example the Projecte Fressa [6] that had development successfully several applications for different kinds of disabilities. One of this programs is the "Mouse Control through voice commands" that allows to control the mouse by means of the voice, it was designed for people that can not control the mouse in a regular form and they could have the control of their voice [6].

In this work we presented an electronic game (memory game) that used the most natural way of human communication: the Speech. This game is designed to be used with children in scholar age that have speech problems. Its characteristics, its environment and its integration with the recognition of voice system are described.

2 Memory Game

In this section we presented the conceptual design of Fonetix system and the memory game, as well as the elements and general characteristics of the game, its history, the interfaces and its characters.

2.1 Conceptual Design

When this project began, we proposed develop applications that were interesting, motivating and that support to children with language problems, for this reason and working co-ordinately with teachers specialized in language area from the Multiple Attention Center (MAC), located in Tizimín, Yucatan, we decided to implement this game, mainly because children enjoy playing it. Once the game had been defined, we didn't want this application was just the game itself, but we wanted to develop a story too, with an environment known by children from this region of the country. Furthermore, it was decided (with the teachers from MAC) to use a collection of direct and indirect words to speech recognition. A direct word is formed by syllables which begin with a consonant, followed by a vowel. An indirect word is formed by at least one syllable that begins with a vowel, followed by a consonant.

The next step we decided that all game story is developed in a house and that the environment from the memory game was a room of the same house. In the same way 4 characters were defined among them, the child could choose and this character the child helps to the character to defeat the challenges that an Alux (mythological character of Mayan culture) presents him during the interaction of the game.

In the Mayan mythology, an Alux is a little spirit that travels through the earth. An Alux is a little taller than an adult person's knee. It wears traditional Mayan clothes; generally is invisible but become into physical form to communicate with human beings.

From the previous, Fonetix system was born. In its first stage the system contains the memorama game. This system recognizes direct and indirect words from child's voice. Actually we are developing two more games, with other collections of words that are used in children's rehabilitation.

2.2 History of Game

At the beginning of the game, an introduction is presented the character that you could chose (Rafa, Tony, Luci or Migue) they have their toys and another articles disordered for all the room, in this moment their mother enters and warns them that if he/she does not order his/her room the Alux will come and take out all his/her things. The character does not obey his/her mom, maybe because he/she does not believe in the Aluxes or maybe because is very lazy, but at the moment the character decides to go out of his/her room but the Alux appears. Seeing the situation, the Alux decides to have fun with the child for a while, and turns every thing in the room into stone. These "stones" fly in the air, disappear and appear like stones in a huacal that is in the room.

Huacal is a term in Mayan language that means a type of cage or box, formed by separated loads of wood, and it is used to transport fruits, vegetables or small animals. After that the Alux informs to the character that if he/she touch 2 stones and them results being objects from the huacal he/she has to pronounce the name of the object, if he/she pronounce correctly the name of the object, this are going to be ordered automatically, but he/she only has 3 chances to pronounce the name of the object. The Alux disappear and leaves the character sad and with a lot of stones inside the huacal. Now is the turn of the child to find the objects

2.3 Description of the Game

Memorama is a game in which the child has to find the pair of stones that contains the same object, touching only two of them in each opportunity. This will let the child to put every object in the correct place in the room. The game has basically two views. The first view is the memorama game itself, represented by the set of stones in the huacal. The another view is the character's room, which contains the untidy objects first, and the animation of the found objects at last.

The game begins with the selection of one of the four characters which the child will use. After that, the Alux's inscription is presented, which turns the object into stones. Once this part has finished, it will be presented to the child the view of the room. This view will be different for each character (Fig. 1). In the upper corner of the view is presented the image of the selected character, as well as the huacal that contains the objects become into stones. The huacal is the link to go to the view of the memorama game.

When you select the image from the huacal, is presented the screen that represent the bottom of the huacal with the stones inside, in the superior part is presented the score of the game (Fig. 2).

To select a stone, the child locates and selects it with a click of the mouse. In this way when the player select a stone this turns in its original form with the name of the object in the lower part, and stayed in this ways until the player select a second stone (Fig. 2). If at any moment, the player wants to see the view of the room, he only selects the small house located in the upper part of the current view. The player can exchange between the two views the times and the moment he wants.

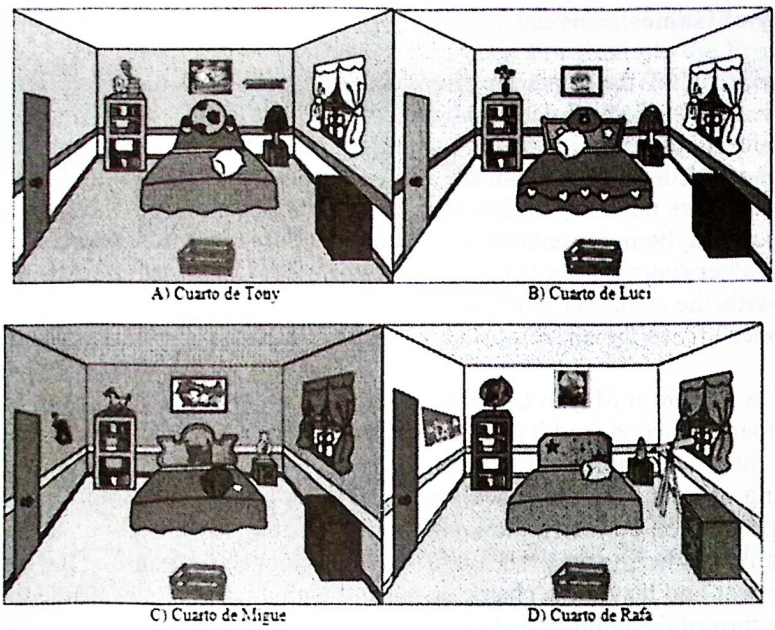


Fig. 1. Images Tony, Luci, Migue and Rafa's rooms.

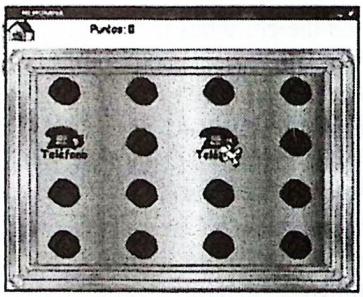


Fig. 2. View thar represents the bottom of the huacal, with the stones inside.

When the player selects a pair of stones, two situations are presented: the first is when the stones selected contain different objects. Then, the objects turn into stones again, hiding their form and the game continues. The second case is when the objects in the stones are the same. When this happens, the system asks the user to pronounce the name of the object correctly, giving him three opportunities to find it. If the option of sound is abled, the system pronounces the word before the user can do it. It is important to say that this option is very useful for a child who doesn't read or those who need to exercise their audition.

If the player can pronounce correctly the name of the object, then the system gives him points: If the word was pronounced correctly in the first opportunity, the system gives 400 points. If it was obtained in the second chance, they are 300 points; in the

last try, only 200 points are regarded. Besides the points obtained, the player can see in the view of the room an animation of the pronounced object (Fig. 3).

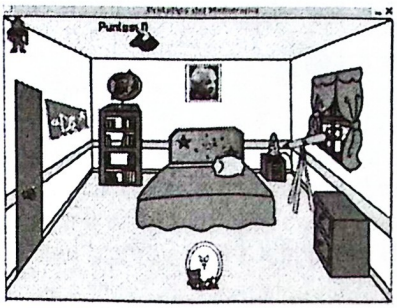


Fig 3. Animation of the correctly pronounced object.

If in the three opportunities the player can't pronounce correctly the name of the object, the system just gives him 100 points and the animation of the object is not presented.

When the objects are the same, the pair of stones selected dissapears from the screen. The game ends when all the objects have been found; that means there are no more stones in the huacal and all the objects are in the correct place in the room. The system shows the room tidied up.

Once the game has finished, a chart is presented to the player, which contains the punctuations, and shows the ten best positions and their times. So, the goal of the game is to tidy the room up, turning the stones into objects, obtaining the better punctuation in the less possible time.

Finally, the player sees a screen where the system gives the opportunity to play the next level or exit the game. It is important to mention that the game has three levels of complex (number of direct and indirect words to be pronounced): in the first level there are 4 words, in the second level there are 8 words, and in the third level there are 12 words.

2.5 Characters

In this section we will present the characters of the game. Their characteristics were designed according to the children's environment, so they are identified with each character. It is important to mention that the names of the characters was put to the vote and were chosen by children that assist to Multiple Attention Center, from Tizimin.

The first of them is Tony (The sportsman): He is 9 years old and he likes a lot sports, he likes to works in teams and he prefer the sports in group, because he has the opportunity to spend time with his friends. His favourite food is stew (Fig. 4).

The second of them is Rafa (The studious): He is 10 years old, he likes reading and he help his partners with their homework. His favorite subjects are math and astronomy, because when he grows up he wants to be an astronaut. His hobbies are: seeing the stars with his telescope and solving riddles (Fig. 5a).



Fig. 4. Tony



Fig. 5. a) Rafa, b) Luci, c) Migue's images

The third of them is Luci (The romantic): she is 8 years old, she likes listening to music and reading poems. lessons and she loves helping her mother baking cookies (Fig. 5b). The last of them is Migue (The naughty): he is Luci's younger brother, he is 6 years old, he likes cartoons and collecting insects. He wants to be a cowboy, like his father. On Sundays, he likes go to the market to eat 2 cochinita tacos and drink horchata (Fig. 5c).

3 Speech Recognition System

The speech recognizing system used 3 main models:

- *A language model* which is based on a finite-states grammar. In this grammar there are defined all the words, direct and indirect, used in the memorama game. With this grammar, 60 words are recognized. These words were defined co-ordinately with the teachers of CAM. Some examples are: *ardilla, aspiradora, espada, escoba, vaso, zapato, dado, corona, cepillo, basura, espejo, hamaca, lupa, goma, galleta, payaso, bate, jirafa, camisa, antifaz, ovni, etc (*These Spanish words are means in English is: *squirrel, vacuum cleaner, spade, broom, glass, shoe, dice, crown, brush, trash, mirror, hammock, gum, cookie, clown, bat, giraffe, shirt, mask, ufo,* respectively*).*
- *A pronunciation model* formed by the necessary vocabulary to define the grammar phrases. The pronuncation is represented as the sequence of phonemes

corresponding to each word. The vocabulary in this game is formed by 60 direct and indirect words.

- *An acoustic model,* this acoustic model was trained with a voice corpus, designed with a linguistic content, which is based on sentences formed with phrases with linguistic meaning that contain the most of the Spanish phonemes from Mexico, this acoustic model was created and used in [7]. The acoustic model was trained with voice corpus formed by 40 people (23 men and 17 women) from different states from the Mexican Republic with mother language is the Spanish talked in Mexico [3]. The phonetic models are independent from the context and identify the phonemes from the Spanish talked in Mexico. For the training of this model were used the libraries from Hidden Markov Tool Kit libraries, HTK [8].

4 Integration between the Electronic Game and Speech Recognition System

The electronic game (memory game) was developed in the Language C and with the SDL graphic libraries. The functions of the speech recognition system was developed in Language C too, these functions contains scripts with HTK's commands. The electronic game with voice recognition runs in Linux.

The interaction between both systems start at the moment to select two stones with equal objects, in that moment the recognition of voice process starts and the player has three chances to pronounce on the correct way the object name. Words are detected with a microphone.

In each try, the function denominated *reconocePalabra* is called. The parameter that it uses is the word that corresponds to the found object name. This function does the next steps:

1. Execute the application *rec_alsa*, which saves the audio imput in a format suitable to HTK. This file is called *palabra.sig*
2. Execute the application *mi_rec_htk.sh*, this function used the command HVite from HTK, the language model, the pronunciation model, the acoustic models and the file *palabra.rec* with the 5 words with most probabilities to do pronounced and their probabilities. Lately a file is generated that only has the 5 recognized words and stored them in a file denominated *palabra2.rec*
3. Execute the application *mi_rec_busca.sh*, this function receive as parameter the object found name. Its homework is to look for the parameter that receive in the file *palabra2.rec*, return 0 or 1 if the parameter was found or not in the file from the recognized words, respectively.
4. The value return *mi_rec_busca.sh* is denied and that value return to the application *reconocePalabra* to the electronic game.

Finally, if the word pronounced corresponds to the object found in the memorama game, the player takes the assigned points.

5 Results

Actually We are freeing up the final version of this electronic game with speech recognition soon, to be used for language-problem children at MAC from Tizimin, Yucatan. Once we have finished this game, we want teachers from MAC uses it from 3 to 4 months continuously to verify its uselfulness in children's rehabilitation. However, we have made proofs with a beta version of the software and the results obtained are this:

- *An nice environment,* children likes the environment of this game; they get excited when they select their own characters. Children are interested in the game. This could be proved when the most 5 mischievous children, according to the their teachers, played the game. They paid a lot of attention in what was happening with the characters in the game.
- *A challenge to defeat,* children made a big effort to pronounce the words correctly; even they talk louder or repeat the words several times because they wanted to see the animation.
- *Satisfactory voice recognition,* this system was proved in children with different levels of language problems, from children with an acceptable pronunciation to children who use brackets or have a lower pronunciation than expected. In the most of the cases the system recognized correctly a lot of words pronounced.
- *A funny and addictive game,* children were excited about using the game; even when they were asked to give the opportunity to other children, they didn't want to stop playing.

This results were obtained in a qualitative way by watching 8 children (of both sex) when they interacted with the game, and in a complementary way with the comments of the 3 teachers of MAC.

6 Conclusions

In this word we described the characteristics, history and characters from an electronic game (memory game) with speech recognition, as well as the way people can interact with this game.

The speech recognition system can recognize the phonemes from Mexican Spanish, These were acoustically modelled using the context-independent modelling, and were trained using Hidden Markov Models. This continues Spanish speech recognition system is multispeaker and it uses a 60 words vocabulary (both direct and indirect words), which is used in children's rehabilitation. We presented the way in which the Memorama game was integrated with the speech-recognizing system, too.

Furthermore, it was presented the preliminary results by using the game in children with language problems. The results obtained are: *the game is funny and addictive, it has an attractive environment, play with it is a challenge to defeat and the voice recognition with children with language problems was satisfactory.* We pretend to corroborate and increase these positive results by doing in the next months quantitative and qualitative evaluations at the moment of using the game in a daily way in the MAC in Tizimin, Yucatan.

Finally, our goal is this software can be a funny, entertaining and less boring tool used to help children to repeat the words in their process of rehabilitation and evaluation. The teachers of CAM say that this tool will be very useful for children.

Acknowledgments. We thanks to CONACYT and Gobierno del Estado de Yucatan all the facilities, supports and resources given to the achiviement of our activities. This project is financed by Fondos Mixtos CONACYT-Gobierno del Estado de Yucatan, code YUC-2004-C03-028.

References

1. http://www.nichcy.org/pubs/spanish/fs11stxt.htm.
2. Miranda-Palma C. and Mayora-Ibarra O. Robotic Remote Navigation by Speech Commands with Automatic Obstacles Detection. Proceedings de IASTED International Conference on Robotics and Applications 2003. Salzburgo, Austria, Junio 2003. pp. 53-57. ISBN: 0-88986-357-1, ISSN: 1027-264X. Editorial ACTA Press..
3. Miranda-Palma C., Mayora-Ibarra O. and Urraga-Serratos E. Interface with speech recognition for the robotic manipulation. Proceedings of Workshop Human-Computer Interaction being, Advances in the Computer Science. ENC 2004, Colima, México, Septiembre 2004 pp.72-78 ISBN: 970-692-170-2.
4. Pineda, L.A., Massé, A., Meza, I., Salas, M., Schwarz, E., Uraga, E. y Villaseñor, L. The DIME project. MICAI 2002: Mexican International Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence. Vol. 2313. pp 166-175. 2002
5. Romero, L., To drive by IIMAS the computer investigation. Published on September 2 of 2002 in Gaceta UNAM. No. 3569. ISBN 0188-5138. http://www.unam.mx/gaceta
6. http://www.xtec.es/~jlagares/f2kesp.htm
7. Miranda-Palma, C., (2003), Navigation Robotic System by Means of Voice Commands with Automatic Detection of Obstacles. Mastery Degree Masteries, ITESM, Campus Cuernavaca, Computer Sciences Mastery degree.
8. Young, S. The HTK Book. Cambridge University Engineering Department, December 2002.

Simulating Lumbar Spine Motion: Parameter Estimation for Realistic Modeling

Ahmed Sameh

Computer Science Department, the American University in Cairo.
113 Kaser Al Aini Street, Egypt

Abstract. In this paper a general framework for modeling and simulation of the dynamic, three-dimensional motion response of the human lumbar-spine is presented. Lumbar vertebrae were modeled as rigid bodies and all other flexible joint structures (i.e. ligaments, cartilage, muscles, and tendons) were modeled collectively as massless springs and dampers. Coupling coefficients, providing additional constraints, were incorporated in the model. Unknown model coefficients (nominally spring, damping and coupling coefficients) were automatically determined by systematically matching the model predictions to spine displacement-time data. A robust parameter optimization module (Monte Carlo routine and genetic algorithm) was developed for this purpose.

1 Introduction

The human spine has over 100 articulating joints, which are comprised of "rigid" structures (vertebral bodies) and "flexible joint structures" (i.e. ligaments, cartilage, muscles, and tendons) that permit complex and coupled motion patterns. Loads and daily activities impart demands on the spine that often lead to spinal dysfunction and pain, particularly in the lumbar region where loads and moments are generally the highest. Development of reliable methods and tools to quantify the force-induced displacement response of the spine is an important first step toward understanding and predicting vertebral mechanical behavior. Without resorting to invasive spine measurements techniques, precise assessment of clinically relevant variables, such as vertebral and intervertebral displacements and stiffness, is very difficult to obtain (Colloca et al., 2001). Mathematical models are therefore often used to quantify the forces and moments acting on the spine. The mechanical response of the spine to externally applied static and dynamic forces is dependent on the complex interaction between the flexible joint structures (FJS) and rigid structures that comprise the function spinal unit. The lumbar spine vertebrae were modeled as five rigid bodies representing vertebral segments L1 to L5. Each vertebra was treated as a rigid-body mass possessing six displacement degrees of freedom: three components of translation and three components of rotation. All other flexible joint structures (i.e. ligaments, disc, muscles, tendons, and cartilage) or FJS were modeled as massless springs (elastic elements) and dampers (viscous elements) constraining the motion of the vertebrae in the six degrees of freedom. The flexible connections to the upper and lower part of the spine (thorax and pelvis) were also represented as spring-dashpot

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 480–489

Received 09/06/07
Accepted 31/08/07
Final version 20/09/07

mechanical elements. Figure 1 shows a schematic diagram of the mechanical equivalence of the lumbar spine and the coordinate system.

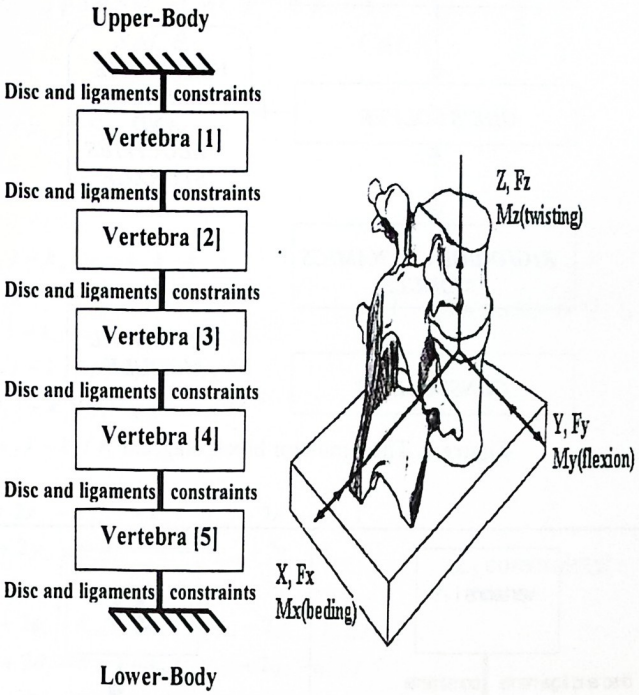


Figure 1. Schematic diagram of the mechanical equivalence of the lumbar spine with description of the coordinate System

Table 1. Vertebrae mass and inertia properties

Property	L_1	L_2	L_3	L_4	L_5
Mass (kg)	0.17	0.17	0.114	0.114	0.114
I_{xx} (10^{-6}) kgm^2	26.7	24.5	16.5	14.8	22.5
I_{yy} (10^{-6}) kgm^2	34.2	31	17.4	20.4	31
I_{zz} (10^{-6}) kgm^2	36.8	36	22.2	26.5	40.3
I_{xz} (10^{-6}) kgm^2	-7.8	-6.9	-2.8	1.4	2.12

Two solvers are implemented in the simulator. For parameter estimations, a first order finite difference procedure is implemented to speed up calculations. For the simulator a fourth order Runge-Kutta procedure is implemented for high accuracy.

$$F_{x_i} - m_i g \sin(\theta_i) = m_i(\dot{u}_i + q_i w_i - r_i v_i)$$

$$F_{y_i} + m_i g \cos(\theta_i)\sin(\phi_i) = m_i(\dot{v}_i + r_i u_i - p_i w_i)$$

$$F_{z_i} + m_i g \cos(\theta_i)\sin(\phi_i) = m_i(\dot{w}_i + p_i v_i - q_i u_i)$$

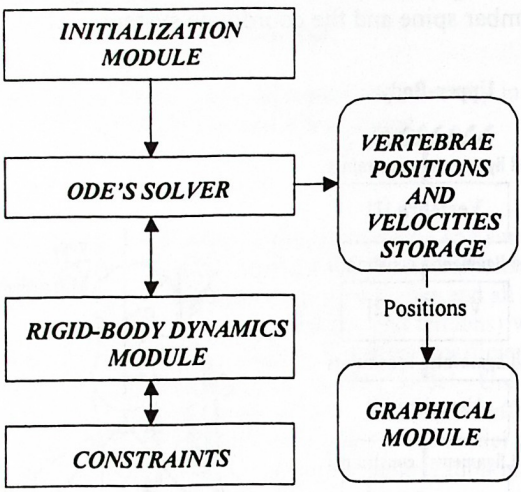


Figure 2. The simulator block diagram

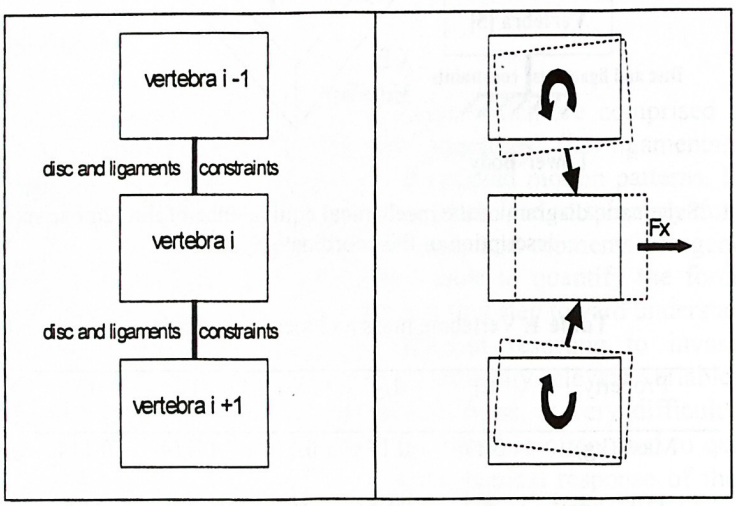


Figure 3. Coupling motion

$$
\begin{aligned}
M_{xi} &= (I_x)_i \dot{p}_i - (I_{xz})_i \dot{r}_i + q_i r_i (I_z - I_y)_i - (I_{xz})_i p_i q_i \\
M_{yi} &= (I_y)_i \dot{q}_i + q_i r_i (I_x - I_z)_i + (I_{xz})_i (p_i^2 - r_i^2) \\
M_{zi} &= \mathrm{n}\,(I_{xz})_i \dot{p}_i + (I_z)_i \dot{r}_i + q_i p_i (I_y - I_z)_i - (I_{xz})_i r_i q_i
\end{aligned}
$$

$$
\begin{aligned}
\dot{\theta}_i &= q_i \cos(\phi_i) - r_i \sin(\phi_i) \\
\dot{\phi}_i &= p_i + q_i \sin(\phi_i) \tan(\theta_i) - r_i \cos(\phi_i) \tan(\theta_i) \\
\dot{\psi}_i &= (q_i \sin(\phi_i) - r_i \cos(\phi_i)) \sec(\theta_i)
\end{aligned}
\qquad (1)
$$

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix}_i = \begin{bmatrix} C\theta_i C\psi_i & S\phi_i S\theta_i C\psi_i - C\phi_i S\psi_i & C\phi_i S\theta_i C\psi_i + S\phi_i S\psi_i \\ C\theta_i S\psi_i & S\phi_i S\theta_i S\psi_i + C\phi_i C\psi_i & C\phi_i S\theta_i S\psi_i - S\phi_i C\psi_i \\ -S\theta_i & S\phi_i C\theta_i & C\phi_i C\theta_i \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

where $Cx_i \equiv cos(x_i)$, $Sx_i \equiv sin(x_i)$

L_1 constraints:

$$\begin{aligned} F_x &= -k_x(x_1 - x_2) - k_u(u_1 - u_2) - k_{x1}x_1 \\ F_y &= -k_y(y_1 - y_2) - k_v(v_1 - v_2) - k_{y1}y_1 \\ F_z &= -k_z(z_1 - z_2) - k_w(w_1 - w_2) - k_{z1}z_1 \\ M_x &= -k_\phi(\phi_1 - \phi_2) - k_p(p_1 - p_2) - k_{\phi 1}\phi_1 \\ M_y &= -k_\theta(\theta_1 - \theta_2) - k_q(q_1 - q_2) - k_{\theta 1}\theta_1 \\ M_z &= -k_\psi(\psi_1 - \psi_2) - k_r(r_1 - r_2) - k_{\psi 1}\psi_1 \end{aligned}$$

L_2 to L_4 constraints (i=2, 3, 4):

$$\begin{aligned} F_x &= -k_x(-x_{i-1} + 2x_i - x_{i+1}) - k_u(-u_{i-1} + 2u_i - u_{i+1}) \\ F_y &= -k_y(-y_{i-1} + 2y_i - y_{i+1}) - k_v(-v_{i-1} + 2v_i - v_{i+1}) \\ F_z &= -k_z(-z_{i-1} + 2z_i - z_{i+1}) - k_w(-w_{i-1} + 2w_i - w_{i+1}) \\ M_x &= -k_\phi(-\phi_{i-1} + 2\phi_i - \phi_{i+1}) - k_p(-p_{i-1} + 2p_i - p_{i+1}) \\ M_y &= -k_\theta(-\theta_{i-1} + 2\theta_i - \theta_{i+1}) - k_q(-q_{i-1} + 2q_i - q_{i+1}) \\ M_z &= -k_\psi(-\psi_{i-1} + 2\psi_i - \psi_{i+1}) - k_r(-r_{i-1} + 2r_i - r_{i+1}) \end{aligned}$$

L_5 constraints:

$$\begin{aligned} F_x &= -k_x(x_5 - x_4) - k_u(u_5 - u_4) - k_{x2}x_5 \\ F_y &= -k_y(y_5 - y_4) - k_v(v_5 - v_4) - k_{y2}y_5 \\ F_z &= -k_z(z_5 - z_4) - k_w(w_5 - w_4) - k_{z2}z_5 \\ M_x &= -k_\phi(\phi_5 - \phi_4) - k_p(p_5 - p_4) - k_{\phi 2}\phi_5 \\ M_y &= -k_\theta(\theta_5 - \theta_4) - k_q(q_5 - q_4) - k_{\theta 2}\theta_5 \\ M_z &= -k_\psi(\psi_5 - \psi_4) - k_r(r_5 - r_4) - k_{\psi 2}\psi_5 \end{aligned} \qquad (2)$$

$$\begin{aligned} F_{zc1} &= -k_{zv}F_{v2} & M_{c1} &= -k_{\theta x}F_{x2} \\ F_{zc2} &= k_{zv}F_{v1} - k_{zv}F_{v3} & M_{c2} &= k_{\theta x}F_{x1} - k_{\theta x}F_{x3} \\ F_{zc3} &= k_{zv}F_{v2} - k_{zv}F_{v4} & M_{c3} &= k_{\theta x}F_{x2} - k_{\theta x}F_{x4} \\ F_{zc4} &= k_{zv}F_{v3} - k_{zv}F_{v5} & M_{c4} &= k_{\theta x}F_{x3} - k_{\theta x}F_{x5} \\ F_{zc5} &= k_{zv}F_{v4} & M_{c5} &= k_{\theta x}F_{x4} \end{aligned} \qquad (3a)$$

$$\begin{aligned} F_{zc1} &= -k_{zy}F_{y2} & L_{c1} &= -k_{\psi y}F_{y2} \\ F_{zc2} &= k_{zy}F_{y1} - k_{zy}F_{y3} & L_{c2} &= k_{\psi y}F_{y1} - k_{\psi y}F_{y3} \\ F_{zc3} &= k_{zy}F_{y2} - k_{zy}F_{y4} & L_{c3} &= k_{\psi y}F_{y2} - k_{\psi y}F_{y4} \\ F_{zc4} &= k_{zy}F_{y3} - k_{zy}F_{y5} & L_{c4} &= k_{\psi y}F_{y3} - k_{\psi y}F_{y5} \\ F_{zc5} &= k_{z1}F_{y4} & L_{c5} &= k_{\psi y}F_{y4} \end{aligned} \qquad (3b)$$

The parameter estimation algorithm proceeds as follows:

1- Initialize parameters (spring and dashpot coefficients).
2- Run the simulator with known input-output data pairs.
3- Calculate the simulator output and compare it with the actual spine output and calculate the error function.
4- Repeat steps 1-3 through the optimizer to minimize the error function till reaching minimum.
5- End.

The optimizer uses a robust optimization procedure based on a Monte Carlo procedure followed by genetic algorithm (GA) detailed in the following section. Parameter estimates for the spring, damper and coupling coefficients are not generally available, or are highly variable. Hence, a Monte Carlo technique (Murray, 1972) was employed to obtain candidate solutions to start a genetic algorithm (GA) (Goldberg, 1989). The Monte Carlo technique implemented in the optimizer is as follows:

1- Generate the unknown parameters randomly.
2- Substitute into the cost function and calculate the error.
3- Repeat step 1 and 2 (n) times.
4- Keep the best m trails with the minimum error.
5- Pass these trials to the GA algorithm.

The GA proceeds with the initial population taken from the Monte Carlo procedure. The full optimizer block diagram is shown in Figure 4.

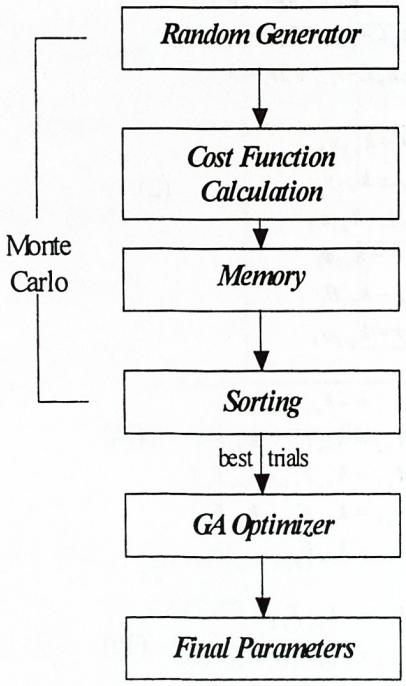


Figure 4. Block diagram of the unknown spring, damper and coupling coefficient parameter estimation procedure

Alternatively, if initial spring, damper and coupling parameters estimates are available, then an optimization technique such as steepest decent is employed. For a single variable, this can be written as

cost function (f) = $\int |e(t)|\,dt$

where $e(t) = V_{exp}(t) - V_{model}(t)$,

For multiple variables, such as displacement-time histories for different axes, the total cost function is just the summation of cost function of each variable with weighting constants as follows:

Total cost function (F) = $w_1 f_1 + w_2 f_2 + ... + w_n f_n =$

$$w_1 \int |e_1(t)|\,dt + w_2 \int |e_2(t)|\,dt + ... + w_n \int |e_n(t)|\,dt$$

Six displacement-time histories data represent the posteroanterior (PA or X-axis) displacement, axial (AX or Z-axis) displacement, and flexion extension (FE) rotation (about Y-axis) for the L3 and L3-L4 vertebral segments subjected to an impulsive force of 100 N applied to the L3 vertebra of a 36 year old, 185 cm, 82 kg male volunteer. Displacement-time histories, with equal time steps (0.2 msec.) were obtained from Keller's model (Keller et al.,2002). The cost function is given by

Total cost function (F) = $\sum_{i=1}^{6} \varepsilon_i$

where

$\varepsilon_1 = \sum |(axial\ \text{displacement}_{Keller} - axial\ \text{displacement}_{Model})|$ for L_3

$\varepsilon_2 = \sum |(axial\ \text{displacement}_{Keller} - axial\ \text{displacement}_{Model})|$ for L_4

$\varepsilon_3 = \sum |(PA\ \text{displacement}_{Keller} - PA\ \text{displacement}_{Model})|$ for L_3

$\varepsilon_4 = \sum |(PA\ \text{displacement}_{Keller} - PA\ \text{displacement}_{Model})|$ for L_4

$\varepsilon_5 = \sum |(FE\ \text{displacement}_{Keller} - FE\ \text{displacement}_{Model})|$ for L_3

$\varepsilon_6 = \sum |(FE\ \text{displacement}_{Keller} - FE\ \text{displacement}_{Model})|$ for L_4

The cost function is given by

Total cost function (F) = $\sum_{i=1}^{3} \varepsilon_i$

where

$\varepsilon_1 = \sum |(axial\ rotation_{Tsung} - axial\ rotation_{Model})|$ for L_3

$\varepsilon_2 = \sum |(lateral\ bending_{Tsung} - lateral\ bending_{Model})|$ for L_3

$\varepsilon_3 = \sum |(flextion\ extension_{Tsung} - flextion\ extension_{Model})|$ for L_3

2 Results

Figure 5 graphically illustrates the experimental (Keller et al., 2002) and parameter optimized axial (AX), transverse (PA), and FE rotation displacement-time histories. Coefficients derived from the impulsive force optimization procedure were used to simulate the PA static and

oscillatory force response (100 N peak amplitude). The model predictions compare favorably with experimental results (Table 2, 3).

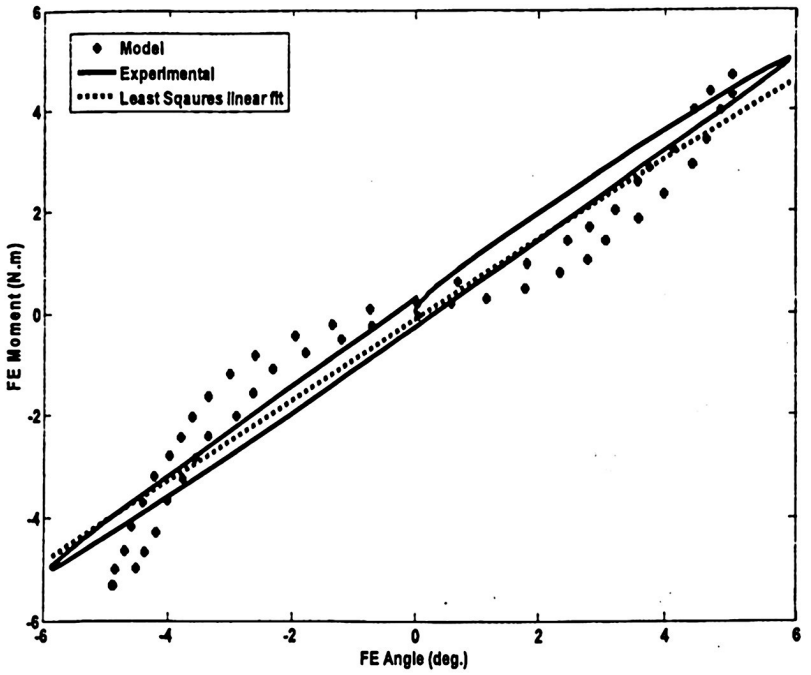


Figure 5. Experimental and parameter optimized axial (AX), transverse (PA), and FE rotation displacement-time histories

Table 2. Peak-to-peak displacements obtained for static PA forces of 100N.

Vertebral Segment	Displacement Axis	Keller's	Spinal Model	% error
L_3	PA (mm)	7.97	7.95	0.2
	AX (mm)	0.131	0.137	-1.9
	FE rotation (°)	0.374	0.355	4.99
L_4	PA (mm)	5.42	5.38	0.3
	AX (mm)	0.70	0.759	-8.6
	FE rotation (°)	1.83	2.03	-10.8

Table 3. PA Peak-to-peak L_3, L_3-L_4 response sinusoidal force at L_3

	2 Hz		5.2 Hz	
	L_3	L_3-L_4	L_3	L_3-L_4
Keller's data [2]	8.23	2.65	7.67	2.55
Spinal model	7.2	2.3	6.1	2
% error	12.2	13.1	19	21

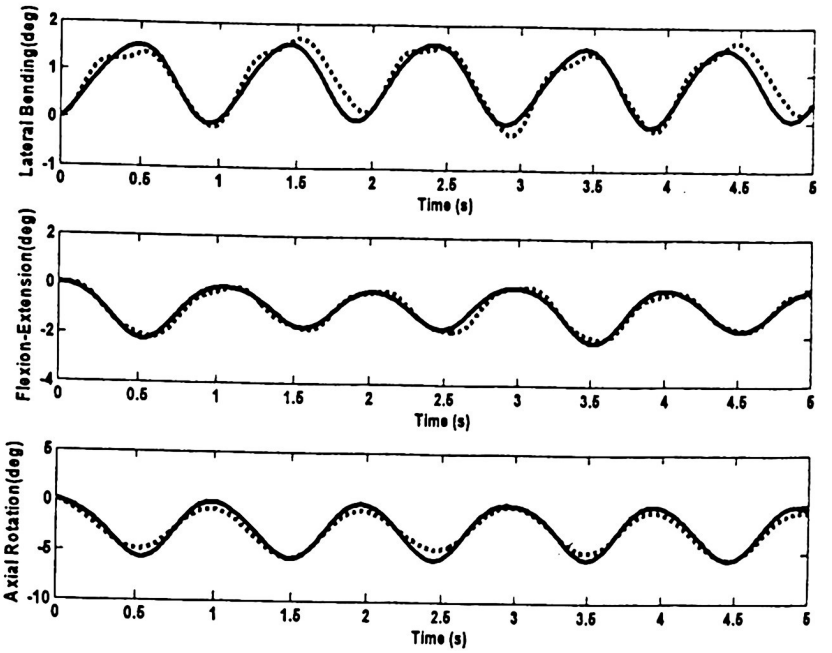
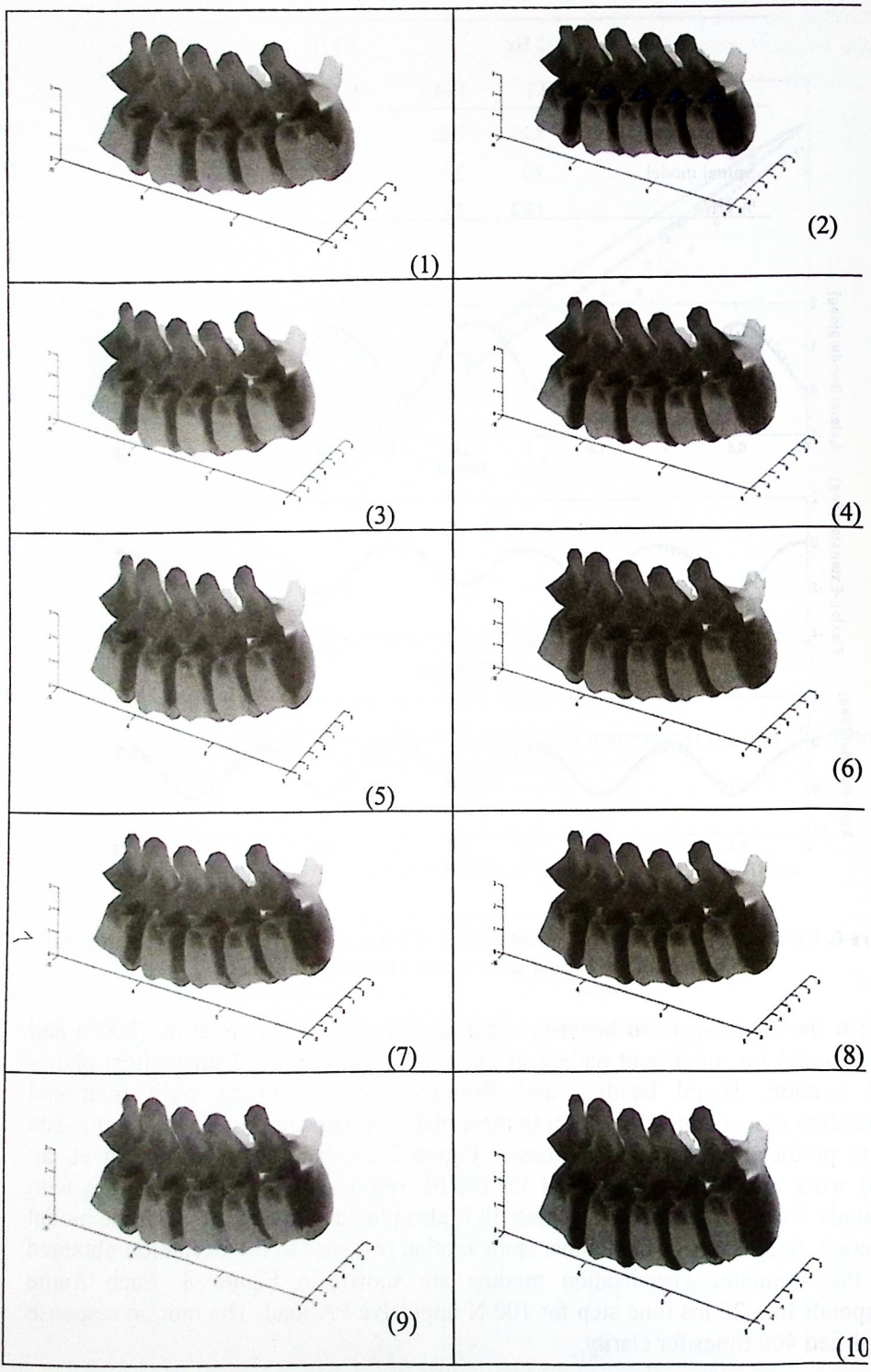


Figure 6. Comparison between experimental work of Evans and current model for movement pattern of the lumbar spine in the 3 anatomical planes

Figure 6 shows comparison between experimental work of Evans et al. (2005) and current model for movement pattern of the lumbar spine in the 3 anatomical planes (axial rotation, lateral bending and flextion-extension) during right rotational mobilization movement with respect to the initial starting positions. These are the data used to predict the rotational stiffness. Figure 7 compares between Ralph et. al. (2005) work and the current model for the FE response for cyclic excitation with magnitude 5N.m . A least-squares linear fit is also ploted which agrees with the model predection. Snap shots of the lumbar spine motion response to PA excitation obtained from the simulator visualization module are shown in Figure 8. Each frame corresponds to a 20 ms time step for 100 N impulsive PA load. The motion response is amplified 400 times for clarity.


(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10

3 Conclusions

The semi-analytical lumbar spine dynamic motion simulator was tested using PA, FE and axial segmental and intersegmental displacement-time data obtained from impulsive force experiments and angular movement pattern from rotational mobilization. Parameter optimized model simulations showed good agreement with the test data, and subsequent independent validation of the static and oscillatory displacement response demonstrated that impulsive force and rotational mobilization test data can be used to predict the lumbar spine motion response during other types of loading conditions. Lumped parameter models, therefore, provide an efficient and effective method to determine the vertebral and/or intervertebral displacement-time history response of the lumbar spine to static, dynamic and impact forces.

References

1. Tony S. Keller, Christopher J. Colloca, Jean-Guy Beliveau, 2002. Force-deformation response of the lumbar spine: a sagittal plane model of posteroanterior manipulation and mobilization, *Clinical Biomechanics*, *17*(3), 185-196.
2. Lee RYW, Evans JH., 1994. Towards a better understanding of posteroanterior mobilization. Physiotherapy 80, 68-73.
3. Lee M., Kelly DW, and Steven, 1995. A model of spine, ribcage, and pelvic responses to a specific lumbar manipulative force in relaxed subjects, Journal of Biomechanics 28(11): 1403-1408.
4. Tsung Bonnie Y. S., Evans JH, Pin Tong, Lee Raymond Y. W., 2005. Measurement of lumbar spine loads and motions during rotational mobilization, Journal of Manipulative and Physiological Therapeutics, vol. 28, no4, pp. 238-244
5. Amed A. Shabana, 2001. Computational Dynamics. John Wiley and Sons, pp. 423-427.
6. Robert C. Nelson, 1998. Flight stability and automatic control. McGraw Hill, pp.
7. Christian P. Robert, George Casella, 2005. Monte Carlo Statistical Methods. Springer. pp. 157-203.
8. D. Goldberg,1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston, MA. pp. 106-120.

Author Index
Índice de autores

Acevedo-Dávila, Jorge 38
Alimi, Mohamed Adel 66
Altamirano Robles, Leopoldo 460
Aranda, Jorge Isidro 27
Aranda, Mary 238
Arrazola, José 89
Baccour, Leila 66
Baltazar, Arturo 27
Barrón-Estrada, M. L. 385, 406
Bravo Sanchez, Ulises Ivan 16
Bulas-Cruz, José 99
Cabrera, Agustín 419
Calvo, Hiram 297, 341
Camal-Uc, Reyna 471
Cao, Yafei 45
Carballido, José Luis 89
Carr, Donald 3
Castillo Borja, Florianne 16
Cen-Magaña, José 471
Chairez, Isaac 419
Chairez, J. 430
Chang, Chingwei 120, 450
Chi, Yenping 120, 450
Contreras-Castillo, Juan 77
Crawford, Broderick 238
Crispín Zavala-Díaz, José 282
Cruz Reyes, Laura 185
Cruz-Chávez, Marco Antonio 282
Cruz-Ramírez, Nicandro 214
Damián-Reyes, Pedro 77
Daniel-Alejandro, García-López 214
Daskalopulu, Aspassia 161
Díaz, R. V. 442
Díaz-Parra, Ocotlán 282
Domenzain Ortega, Lourdes 269
Dominguez, Elena Rosa 442
Dong, Hong-zhao 258
Errecalde, Marcelo 150
Escalante, Manuel 471
Fang, Kwoting 120, 450
Favela, Jesús 77
Fernandez, Luis M. 193
Ferretti, Edgardo 150
Galindo Domínguez, M. Coral 460
Garcia, Alejandro 150
García, Zenaida 442
Giannikis, Georgios K. 161
Gómez de Silva Garza, Andrés 269
Gómez Santillán, Claudia 185
Gómez-Balderas, Ernesto 305
González, Jesús A. 460
Gonzalez-Segura, Cinhtia 471
Gonzalez-Segura, Sergio 471
Guha Thakurta, Abhradeep 363
Guo, Ming-fei 258
Hashimoto, Kazuo 175
Héctor-Gabrie, Acosta-Mesa 214
Hernandez, Angeles 204
Hu, Daiping 141
Hu, Jie 111
Iglesias Silva, Gustavo Arturo 16
Jiang, Peilin 374
Jiménez-Salazar, Héctor 305
Juárez-Romero, David 282
Kanoun, Slim 66
Khan, Hashir 363
Krömer, Pavel 56
Kubo, Shota 175
Kumar K., Ashwath 363
Kuroiwa, Shingo 330, 374
Lezcano, Mateo 442
Li, Mengyu 141
Li, Ran 374
Li, Sikun 45
Li, Zhuoming 330
Maghrebi, Wafa 66
Marreiros, Goreti 99
Martínez-Rangel, Martín G. 282
Mendez, Gerardo M. 204
Méndez-Ortega, Luis Andrés 131
Meza Conde, Eustorgio 185
Miranda, Ramón 419
Miranda-Palma, Carlos 471
Monfroy, Eric 227

Neves, José 99
Olarte, Carlos 227
Olmos Pineda, Ivan 460
Ono, Ryuta 175
Ortega Izaguirre, Rogelio 185
Ortega-Pacheco, David 341
Osorio, Mauricio 89
Osorio-Velazquez, J. Moises 406
Ouddane, Nabil 56
Parama, Jose R. 352
Paredes-Farrera, Mayra A. 297
Pedreira, Oscar 352
Peng, Yinghong 111
Pérez-Molina, Clara 395
Pulido, JRG 77
R. Brisaboa, Nieves 352
Ramos, Carlos 99
Ren, Fuji 330, 374
Reyes-Galaviz, Orion Fausto 131
Reyes-García, Carlos A. 385, 406
Rico Martinez, Ramiro 16
Ríos, Jorge 471
Rodríguez, A. 430
Rodríguez-Artacho, Miguel 395
Rotstein, Nicolas 150
Rueda, Camilo 227
Ruelas, Ruben 3
S. Places, Angeles 352
Salgado, I. 430
Sameh, Ahmed 480
Sandoval-Sánchez, Guillermo A. 385
Santillan, P. 430
Santos, Ricardo 99
Seco, Diego 352
Sidorov, Grigori 305
Simari, Guillermo R. 150
Snasel, Vaclav 56
Soo, Von-Wun 315
Terabe, Masahiro 175
Terashima-Marín, Hugo 193
Torres Treviño, Luis 38, 249
Turrubiates López, Tania 185
Valenzuela-Rendón, Manuel 193
Vega-Zazueta, L. 385
Wang, Dawei 45
Wang, Qi 141
Wu, Ruiming 141
Xiao-hong, Qian 258
Xiong, Guangleng 111
Yang, Shih-Yao 315
Zatarain, Ramón 385
Zatarain-Cabada, Ramon 406
Zepeda-Sanchez, L. 406
Zhang, Yu 330
Zheng, Nanning 374
Zhou, Min 258
Zúñiga, A. 430

Editorial Board of the Volume

Comité editorial de volumen

Ajith Abraham
Mikhail Alexandrov
Gustavo Arroyo
Ildar Batyrshin
Bedrich Benes
Igor Bolshakov
Paul Brna
Andre C. P. L. F. de Carvalho
Hiram Calvo
Nicoletta Calzolari
María José Castro Bleda
Simon Colton
Ulises Cortes
Nareli Cruz-Cortes
Louise Dennis
Juergen Dix
Abdennour El Rhalibi
Denis Filatov
José Galaviz
Sofía N. Galicia-Haro
Matjaž Gams
Alexander Gelbukh (Co-chair)
Arturo Hernández-Aguirre
Jesse Hoey
Dieter Hutter
Pablo H. Ibargüengoytia
Ryszard Klempous
Mario Köppen
Angel Kuri (Co-chair)
Ana Lilia Laureano-Cruces
Steve Legrand
Christian Lemaître León
Eugene Levner
James Little
Aurelio López
Jacek Malec
Efren Mezura-Montes
Mikhail Mikhailov
Chilukuri Mohan
Raúl Monroy
Eduardo Morales
Guillermo Morales Luna
Juan Arturo Nolazco Flores
Mauricio Osorio Galindo
Manuel Palomar
Oleksiy Pogrebnyak
Fuji Ren
Carlos Alberto Reyes-García
Riccardo Rosati
Paolo Rosso
Stefano Rovetta
Khalid Saeed
Andrea Schaerf
Leonid Sheremetov
Grigori Sidorov
Humberto Sossa Azuela
Benno Stein
Thomas Stuetzle
Luis Sucar
Hugo Terashima
Berend Jan van der Zwaag
Javier Vázquez-Salceda
Manuel Vilares Ferro
Luis Villaseñor-Pineda
Toby Walsh
Alfredo Weitzenfeld
Franz Wotawa
Kaori Yoshida
Carlos Mario Zapata Jaramillo

Additional Reviewers

Árbitros adicionales

Mohamed Abdel Fattah
Omar Abuelma'atti
Luis Alberto Pineda
Jose Arrazola
Héctor Avilés
Alejandra Barrera

Lucia Barron
Marcio Basgalupp
Mike Baskett
Tristan Behrens
Edgard Benítez-Guerrero
Edmundo Bonilla
Matthew Brisbin
Nils Bulling
Sara Carrera Carrera
Oscar Celma
Zenon Chaczko
Marco Chiarandini
Victor Manuel Darriba Bilbao
Rogelio Davila Perez
Yogesh Deshpande
Joseph Devereux
Luca Di Gaspero
Pantelis Elinas
Cora Beatriz Excelente Toledo
Katti Faceli
Bruno Feres de Souza
Milagros Fernández-Gavilanes
Michael Francis
Peter Funk
Karen Azurim Garcia Gamboa
René A. García Hernández
Paola Garcia-Perera
Sergio Gomez
Fabio A. Gonzalez
John Haggerty
Emmanuel Hebrard
Abir Hussain
Diana Inkpen
Ruben Izquierdo-Bevia
Bartosz Jablonski
Héctor Jiménez Salazar
Peilin Jiang
Phil Kilby
Jerzy Kotowski
Zornitsa Kozareva
Ricardo Landa Becerra
Yulia Ledeneva
Domenico Lembo
Agustin Leon
Erica Lloves Calviño
Juan Carlos Lopez Pimentel
Jose Luis Carballido
Michael Maher
Antonio Marin Hernandez
Manuel Mejia
Carlos Mex-Perera
Sven Meyer zu Eissen
Rada Mihalcea
Miguel Angel Molinero Alvarez
Manuel Montes-y-Gómez
Rafael Morales
Paloma Moreda Pozo
Boris Motik
Rafael Murrieta
Mariá Nascimento
Juan Antonio Navarro
Juan Carlos Nieves
Peter Novak
Slawomir Nowaczyk
Constantin Orasan
Ryan Pedela
Bernhard Peischl
Alejandro Peña Ayala
David Pinto
Antonella Poggi
Martin Potthast
Pascal Poupart
Pilar Pozos
Jakob Puchinger
Orion Fausto Reyes-Galaviz
Francisco Ribadas-Pena
Andrea Roli
Israel Román
Marco Ruzzi
William Sellers
Eduardo Spinosa
Geof Staniford
Gerald Steinbauer
Ewa Szlachcic
Sulema Torres
Gregorio Toscano-Pulido
Victor Trevino
Dan Tufis
Javier Vazquez
Andrew Verden
Joerg Weber
Ning Xiong
Fernando Zacarias-Flores
Ramon Zatarain
Claudia Zepeda Cortes

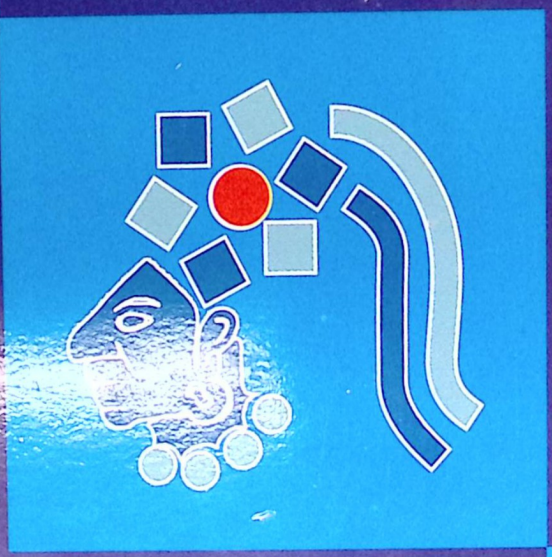
Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2007.
Printing 500 / *Edición 500 ejemplares.*

Artificial Intelligence is a branch of computer science that develops heuristic methods for solving super-complex problems that cannot be solved by direct calculations or mathematics methods, as well as ways to model human intellectual activity.

This volume presents 45 original peer-reviewed research papers on theory and applications of Artificial Intelligence. The volume is structured into seven thematic fields:

- Computational intelligence
- Knowledge representation and reasoning
- Machine learning
- Constraint programming, planning, and scheduling
- Natural language processing
- Intelligent tutoring systems
- Intelligent biomedical applications

The volume is prepared in collaboration with Mexican Artificial Intelligence Society (SMIA).



ISSN: 1870-4069
www.cic.ipn.mx/rcs
www.ipn.mx

ESTADOS UNIDOS MEXICANOS
SEP

SMIA

IPN
INSTITUTO POLITÉCNICO NACIONAL
"La Técnica al Servicio de la Patria"