Knowledge Discovery for Knowledge Based Systems. Some Experimental Results

Rancan C., Kogan A., Pesado P. and García-Martínez R.

Computer Science Doctorate Program. Computer Sc. School. La Plata National University
Software and Knowledge Engineering Center. Postgraduate School. ITBA
Intelligent Systems Laboratory. Engineering School. University of Buenos Aires.
Instituto de Investigación en Informática LIDI, Facultad de Informática, UNLP - CIC Bs As

claudioran@yahoo.com, ppesado@lidi.info.unlp.edu.ar, rgm@itba.edu.ar

Abstract. This paper addresses some considerations based on the state of the involved technologies for the integration of knowledge discovery systems and knowledge based systems centered in automatic knowledge acquisition for experts systems. Some experimental results related to the quality of the generated knowledge bases are shown.

1 Introduction

The knowledge based systems (KBS) or expert systems emulate the human expert behavior in a certain knowledge area. They constitute aid systems to take decisions in different areas such as educational strategic selection [1], environmental variables control [2], neonatology fans configuration [3], agreement in judicial process [4] or the attended generation of activity maps of software development projects [5]. Knowledge based systems to aid decision taking is a one particular knowledge based system.[6]. The knowledge base of an expert system encapsulates in some representation formalism (rules, frames, semantic nets among other), the domain knowledge that should be used by the system to solve a certain problem. The development methodologies of knowledge bases have been consolidated in the last 15 years [7], [8]. The intelligent systems constitute the computer science field which studies and develops algorithms that implement the different learning models and their application to practical problems resolution. Among the problems approached in this field, we can find the one related to knowledge discovering [9]. Knowledge discovery (KD) consists on the search of interesting patterns and important regularities in big information bases [10]. When speaking of knowledge discovery based on intelligent systems or Data/Information Intelligent Mining we refer specifically to the application of machine learning methods or other similar methods, to discover and to enumerate patterns present in this information. One of knowledge discovery paradigms is centered in the knowledge evaluation [11], its structure [12], the distributed acquisition processes [13] and the intelligent systems technologies associated to the knowledge discovery [14]. The interaction between knowledge based systems and discovery systems has antecedents in the paradigm of integrated

© S. Torres, I. López, H. Calvo. (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 3-13

Received 09/02/07
Accepted 08/04/07
Final version 20/04/07

architectures of planning and learning based on theories construction [15] and hybrid architectures of learning [16], [17], [18]. In this context, this paper introduces the problem (section 2), an integrative proposal is formulated (section 3), components are identified (section 3.1) and the interaction between them (section 3.2), an example is provided that illustrates partially how the workspace would work (section 4), some experimental results are shown (section 5), finally related work (section 6) future research are adressed (section 7).

2 Problem

Recent works in decision making systems in strategic – operational workspace based on KBS like air control or naval units readiness areas [19] show that it is an open problem to define how KBS can be integrated to knowledge discovery processes based on machine learning that allow them to improve "on-line" the quality of the knowledge base used for decision making. Approaches for solving this type of problem are addressed for incremental improvement of decision making systems in office automation area [20].

3 Toward an Integrative Proposal

In this section the components of the integrative proposal are presented (section 3.1) and the interactions between these components (section 3.2).

3.1 Identification of the Components

3.1.1 The Bases

This section describes: the knowledge base, the concepts dictionary, the examples base, the records base, the clustered records base, the clustered/classification rules base, the discovered rules base and the updated knowledge base.

Knowledge Base. This base contains the problem domain knowledge deduced by the knowledge engineer, which contributes the knowledge pieces (rules) applicable to the resolution of the problem outlined by the user of the system.
Concepts Dictionary. This base stores the registration of all the concepts used in the different knowledge pieces (rules) that integrate the Knowledge Base. For each concept it keeps registration of the corresponding attributes and the possible values of each attribute
Examples Base. This base keeps examples of elements that belong to different classes. The attributes of these examples should keep correlativity or should be coordinated with the attributes of the concepts described in the Concepts Dictionary.
Records Base. This base keeps homogeneous records of information which is associated to some process of knowledge discovery. (I/E clustering).

Clustered Records Base. This base keeps homogeneous records of information which are clustered in classes without labeling (clusters) as a result of applying the clustering process to the Records Base.

Clustering/Classification Rules Base. This base keeps knowledge pieces (rules) discovered automatically as a result of applying the induction process to the Clustered Records Base and the Examples Base

Discovered Rules Base. This base keeps knowledge pieces (rules) related to the problem domain as result of applying the labeling conceptual process to the discovered knowledge pieces (rules) that are stored in the Clustering/Classification Rules Base.

Updated Knowledge Base. This base encapsulates the knowledge that becomes from the integration of the problem domain knowledge pieces (rules) educed by the knowledge engineer and the knowledge pieces (rules) discovered automatically as a result of the application of the processes of clustering/induction to the Records Base or induction to the Examples Base.

3.1.2 The Processes

This section describes the processes: cluster, Inducer, conceptual labeler, knowledge integrator and inference engine.

Cluster. This process is based in the use of self organized maps (SOM) to generate groups of records that are in the Records Base. These groups are stored in the Clustered Records Base.

Inducer. This process is based in the use of induction algorithms to generate clustering rules beginning from the records groups that are in the Clustered Records Base and Classification Rules beginning from the records that are in the Examples Base.

Conceptual Labeler. This process is based on the use of the Concepts Dictionary and the Clustering/Classification Rules Base to generate the Discovered Rules Base. This process transforms the knowledge pieces obtained into pieces of coordinated knowledge with the Knowledge Base.

Knowledge Integrator. This process generates the Updated Knowledge Base from the Discovered Rules Base and the Knowledge Base, solving all the integration problems between them.

Inference Engine. It is the process that automates the reasoning to solve the problem outlined by the user, beginning from the pieces of knowledge available in the Updated Knowledge Base or Knowledge Base.

3.2 Interaction among Components

The interaction among the different components is shown in Figure 1. The Knowledge Base encapsulates the necessary pieces of knowledge (rules) for the resolution of domain problems. This interaction with the inference engine constitutes the Knowledge Based System (Expert System). Beginning from the concepts / attributes / values that are present in the different pieces of knowledge inside the

Knowledge Base, the Concepts Dictionary is built. When a situation of knowledge discovery takes place because the Inducer generated a Clustering/Classification Rules Base, or because this has became from an Examples Base or a Clustered Records Base resultanting of applying the Cluster to a Records Base, the pieces of knowledge (rules) that are in the Clustering/Classification Rules Base can present the characteristic of not being coordinated with the available pieces of knowledge in the Knowledge Base.

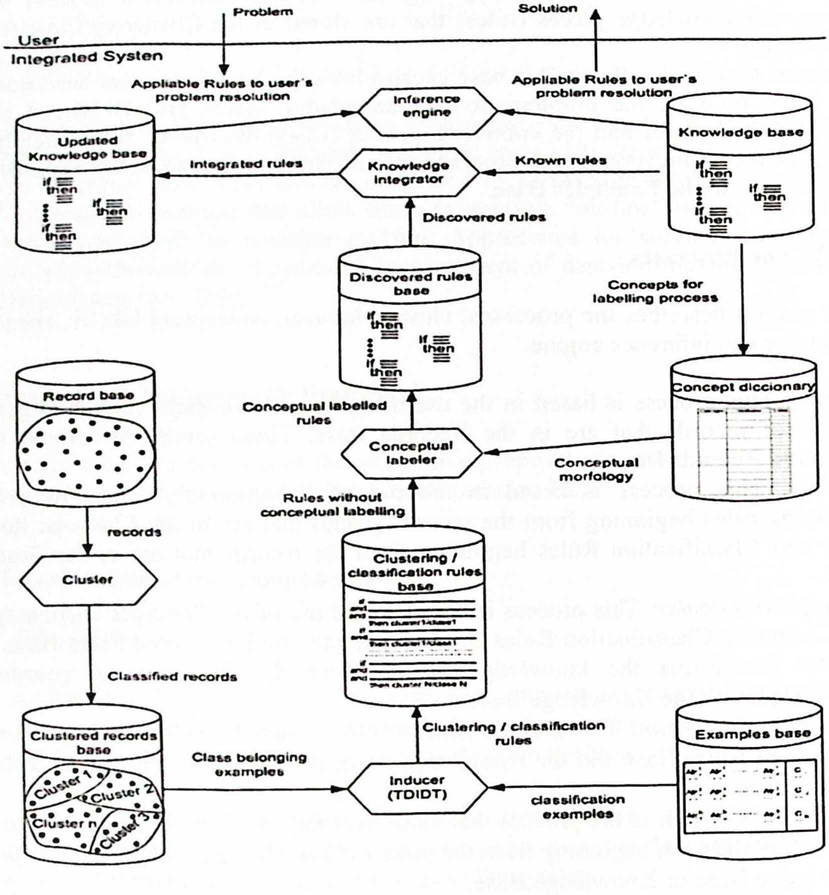

Fig. 1. Interaction among different components

In this context the Conceptual Labeler transforms the knowledge pieces of the Clustering/Classification Rules Base into coordinated knowledge pieces with those rule corresponding to the Knowledge Base generating the Discovered Rules Base. The Knowledge Integrator takes the Discovered Rules Base and (solving the emergent integration problems) integrates it into the Knowledge Base, generating the Updated Knowledge Base, that becomes the new Knowledge Base and the cycle is restarted.

4 An Example in the Ship Operations Cost Domain

Let us consider, for example, the operation costs establishment problem in a ships owner company in function of the ship type to operate in a certain port. Consider the Knowledge Base whose rules are exemplified in table 1. Consider the Concepts Dictionary associated to this Knowledge Base shown in the table 2.

From the Examples Base the Inducer generates the Classification Rules Base shown in the table 4. The Conceptual Labeler identifies the belonging of values to the domain of attributes in Concepts Dictionary generating the Discovered Rules Base shown in the table 5.

The Knowledge Integrator analyzes the Discovered Rules Base, verifying that there are no integration conflicts and proceeds to integrate it to the Knowledge Base generating the Updated Knowledge Base shown in the Table 6. This last one becomes the new Knowledge Base.

Table 1. Knowledge Base

	Rules		Rules
IF	SHIP.SHIP_TYPE= BULK CARRIER	IF	SHIP.SHIP_TYPE= CONTAINER
AND	SHIP.SIZE= LARGE	AND	SHIP.SIZE= LARGE
AND	PORT.PORT_FACILITIES= VERY GOOD	AND	PORT- PORT_FACILITIES= V. GOOD
AND	PORT.ACCESSS= FREEWAY	AND	PORT.ACCESSS; FREEWAY
	THEN COSTS.PIER_LONG= ENLARGE	THEN	COSTS.PIER_LONG= NORMAL
AND	COSTS.MOORING_TIME= HABITUAL	AND	COSTS.MOORING_TIME; SHORT
IF	SHIP.SHIP_TYPE= BULK CARRIER	IF	SHIP.SHIP_TYPE= CONTAINER
AND	SHIP.SIZE= MEDIUM	AND	SHIP. SIZE= MEDIUM
AND	PORT.PORT_FACILITIES= VERY GOOD	AND	PORT.PORT_FACILITIES= VERY GOOD
AND	PORT.ACCESS; FREEWAY	AND	PORT.ACCESSS= FREEWAY
THEN	COSTS.PIER_LONG= ENLARGE	THEN	COSTS.PIER_LONG= NORMAL
AND	COSTS.MOORING_TIME= HABITUAL	AND	COSTS.MOORING_TIME= SHORT
IF	SHIP.SHIP_TYPE= BULK CARRIER	IF	SHIP.SHIP_TYPE= CONTAINER
AND	SHIP.SIZE= SMALL	AND	SHIP.SIZE= SMALL
AND	PORT.PORT_FACILITIES; VERY GOOD	AND	PORT.PORT_FACILITIES= VERY GOOD
AND	ACCESSS= FREEWAY	AND	PORT.ACCESSS= FREEWAY
THEN	COSTS.PIER_LONG= NORMAL	THEN	COSTS.PIER_LONG= NORMAL
AND	COSTS MOORING_TIME= SHORT	AND	COSTS.MOORING_TIME= SHORT
IF	SHIP.SHIP_TYPE= TANKER	IF	SHIP.SHIP_TYPE= PASENGER
AND	SHIP.SIZE= LARGE	AND	SHIP.SIZE= LARGE
AND	PORT.PORT_FACILITIES= VERY GOOD	AND	PORT.PORT_FACILITIES= VERY GOOD
AND	PORT.ACCESSS; FREEWAY	AND	PORT.ACCESS= FREEWAY
THEN	COSTS.PIER_LONG= NORMAL	THEN	COSTS.PIER_LONG= REDUCED
AND	COSTS.MOORING_TIME; HABITUAL	AND	COSTS.MOORING_TIME= HABITUAL
IF	SHIP.SHIP_TYPE= TANKER	IF	SHIP.SHIP_TYPE= PASENGER
AND	SHIP.SIZE= MEDIUM	AND	SHIP.SIZE= MEDIUM
AND	PORT.PORT_FACILITIES= VERY GOOD	AND	PORT.PORT_FACILITIES= VERY GOOD
AND	PORT.ACCESSS= FREEWAY	AND	PORT.ACCESS= FREEWAY
THEN	COSTS.PIER_LONG= NORMAL	THEN	COSTS.PIER_LONG= REDUCED
AND	COSTS.MOORING_TIME= HABITUAL	AND	COSTS.MOORING_TIME= HABITUAL
IF	SHIP.SHIP_TYPE= TANKER	IF	SHIP.SHIP_TYPE= PASENGER
AND	SHIP.SIZE= SMALL	AND	SHIP.SIZE= SHORT
AND	PORT.PORT_FACILITIES= VERY GOOD	AND	PORT.PORT_FACILITIES= VERY GOOD
AND	PORT.ACCESSS= FREEWAY	AND	PORT.ACCESS= FREEWAY
THEN	COSTS.PIER_LONG= NORMAL	THEN	COSTS.PIER_LONG= NORMAL
AND	COTS.PORT.MOORING_TIME= SHORT	AND	COSTS.MOORING_TIME= SHORT

Table 2. Dictionary of Concepts

Concept	Attribute	Value
SHIP	SHIP_TYPE	BULK CARRIER CONTAINER TANKER PASSENGER
	SIZE	SMALL MEDIUM LARGE
PORT	PORT_FACILITIES	VERY GOOD GOOD REGULAR POOR
	ACCESSS	FREEWAY ROUTE ROAD TRACK
COSTS	PIER_LONG	REDUCED NORMAL ENLARGE
	MOORING_TIME	SHORT HABITUAL EXTEND

Table 3. Examples Base

SHIP_ TYPE	SIZE	PORT_ FAC	ACCESSS	PIER_ LONG	MOORING_ TIME
Bulk Carrier	Large	Very Good	Freeway	Enlarge	Habitual
Bulk Carrier	Medium	Very Good	Freeway	Enlarge	Habitual
Bulk Carrier	Small	Very Good	Freeway	Enlarge	Short
Tanker	Large	Very Good	Freeway	Normal	Habitual
Tanker	Medium	Very Good	Route	Normal	Habitual
Tanker	Small	Very Good	Road	Normal	Short
Container	Large	Very Good	Freeway	Normal	Short
Container	Medium	Very Good	Freeway	Normal	Short
Container	Small	Very Good	Freeway	Normal	Short
Passenger	Large	Very Good	Freeway	Normal	Habitual
Passenger	Medium	Very Good	Freeway	Reduced	Habitual
Passenger	Small	Very Good	Freeway	Reduced	Short

Table 4. Classification Rules Base

Rules	
IF THEN	SHIP_TYPE= CONTAINER MOORING_TIME= SHORT
IF THEN:	SHIP_TYPE= CONTAINER PIER_LONG= NORMAL
IF THEN	SHIP_TYPE= BULK CARRIER PIER_LONG= ENLARGE

Table 5. Discovered Rules Base

Rules	
IF THEN	SHIP SHIP_TYPE= CONTAINER COSTS MOORING_TIME= SHORT
IF THEN	SHIP SHIP_TYPE= CONTAINER COSTS PIER_LONG= NORMAL
IF THEN	SHIP SHIP_TYPE= BULK CARRIER COSTS PIER_LONG= ENLARGE

5 Some Experiments

The improvement of a Knowledge Base with discovered knowledge pieces in automatic way can lead to a degradation of the original Knowledge Base, so it is necessary to explore (theoretically at least) which are the curves of degradation of the quality process of knowledge discovery identifying border conditions for the model in the developed theoretical frame. In order to this a three step experiment which structure is shown in figure 2 has been carry out.

Table 6. Updated Knowledge Base

Rules	Rules	Rules
IF SHIP.SHIP_TYPE= BULK CARRIER AND SHIP.SIZE= LARGE AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= ENLARGE AND COSTS.MOORING_TIME= HABITUAL	IF SHIP.SHIP_TYPE= TANKER AND SHIP.SIZE= SMALL AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COTS.PORT.MOORING_TIME= SHORT	IF SHIP.SHIP_TYPE= PASENGER AND SHIP.SIZE= MEDIUM AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESS= FREEWAY THEN COSTS.PIER_LONG= REDUCED AND COSTS.MOORING_TIME= HABITUAL
IF SHIP.SHIP_TYPE= BULK CARRIER AND SHIP.SIZE= MEDIUM AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESS=FREEWAY THEN COSTS.PIER_LONG= ENLARGE AND COSTS.MOORING_TIME= HABITUAL	IF SHIP.SHIP_TYPE= CONTAINER AND SHIP.SIZE= LARGE AND PORT- PORT_FACILITIES= V. GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS.MOORING_TIME= SHORT	IF SHIP.SHIP_TYPE= PASENGER AND SHIP.SIZE= SHORT AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS.MOORING_TIME= SHORT
IF SHIP.SHIP_TYPE= BULK CARRIER AND SHIP.SIZE= SMALL AND PORT.PORT_FACILITIES=VERY GOOD AND ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS MOORING_TIME= SHORT	IF SHIP.SHIP_TYPE= CONTAINER AND SHIP. SIZE= MEDIUM AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS.MOORING_TIME= SHORT	IF SHIP.SHIP_TYPE = CONTAINER THEN COSTS.MOORING_TIME= SHORT
IF SHIP.SHIP_TYPE= TANKER AND SHIP.SIZE= LARGE AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS.MOORING_TIME= HABITUAL	IF SHIP.SHIP_TYPE= CONTAINER AND SHIP.SIZE= SMALL AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS.MOORING_TIME= SHORT	IF SHIP.SHIP_TYPE= CONTAINER THEN COSTS.PIER_LONG= NORMAL
IF SHIP.SHIP_TYPE= TANKER AND SHIP.SIZE= MEDIUM AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESSS= FREEWAY THEN COSTS.PIER_LONG= NORMAL AND COSTS.MOORING_TIME= HABITUAL	IF SHIP.SHIP_TYPE= PASENGER AND SHIP.SIZE= LARGE AND PORT.PORT_FACILITIES= VERY GOOD AND PORT.ACCESS= FREEWAY THEN COSTS.PIER_LONG= REDUCED AND COSTS.MOORING_TIME= HABITUAL	IF SHIP.SHIP_TYPE= BULK CARRIER THEN COSTS.PIER_LONG= ENLARGE

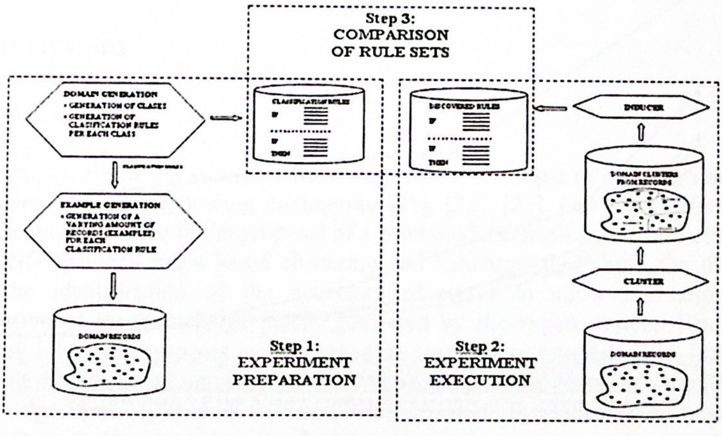

Fig. 2. Structure of he three step experiment

The step 1 consists in experiment preparation. This step involves: [a] domain generation based on: generation of classes and generation of classification rules for each class and [b] examples generation for each classification rule. The output of this step is a classification rules set and a domain records (examples) set. The step 2 consists in experiment execution. This step involves: [a] to apply the cluster process to domain records (examples) set to obtain the domain clusters set and [b] to apply the inducer process to the domain clusters set to obtain the discovered rules set. The step 3 consists on the comparison of the classification rule set from step 1 with the discovered rules set from step 2 the percentage of matching rules defines the experiment success.

5.1 Variables

The experimentation use the following independent variables: [a] *attributes number*: amount of attributes in each classification rule (the same in the examples), [b] *rules per class*: amount of classification rules for determining each domain class, [c] *class possible values*: amount of domain different classes; and the following dependent variable: [b] *rules correctly covered*: percentage of matching rules among classification rules set and discovered rules set.

5.2 Results

The experiments explore the behavior of the processes in domains where classes have associated different amounts of classification rules and the amount of attributes per classification rule can vary and in domains where amount of classes can vary and each class has associated classification rules in which amount of attributes per classification rule can vary. Results of the experiments are shown in figures 3 and 4.

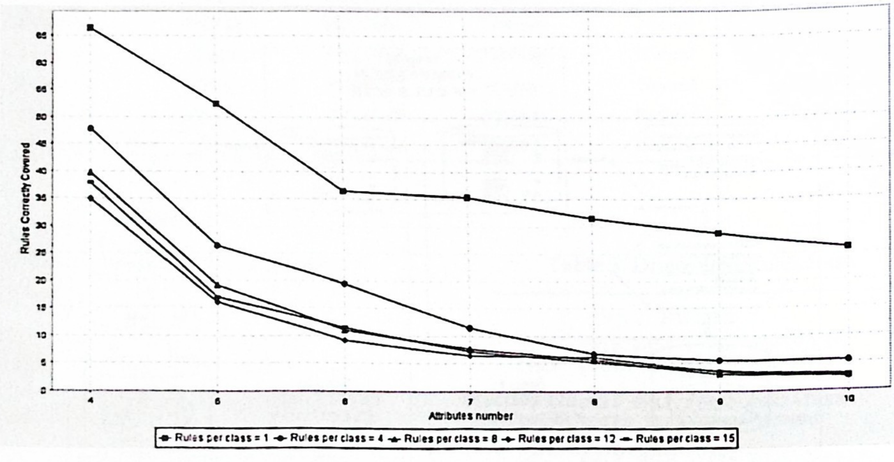

Fig. 3. Domains where classes have associated different amounts of classification rules and the amount of attributes per classification rule can vary.

Figure 1 shows that when domain is complex in terms of amount of attributes needed for classifying (more attributes in a classification rule) or when domain is complex in terms of amount of classification rules needed for identifying a class, the performance (classification rules correctly predicted) of the proposed method (clustering + induction) decreases.

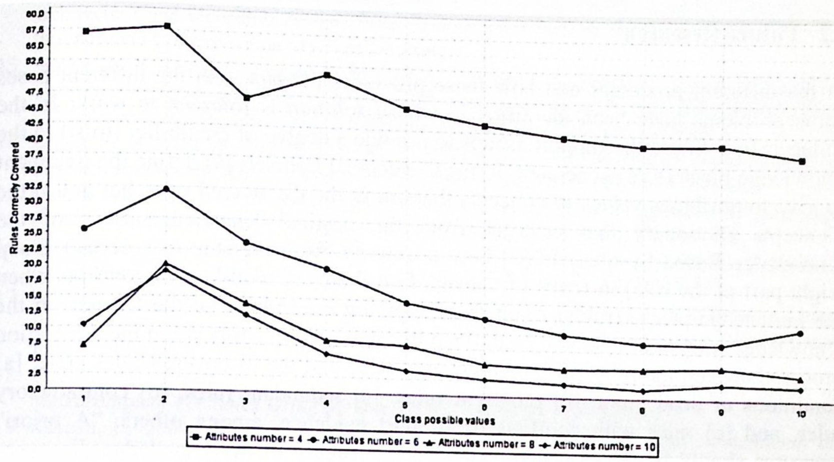

Fig. 4. Domains where amount of classes can vary and each class has associated classification rules in which amount of attributes per classification rule can vary.

Figure 2 shows that when domain is complex in terms of amount of classes the performance (classification rules correctly predicted) of the proposed method (clustering + induction) decreases. Also shows that when the amount of attributes per classification rule for each class decrease, the performance of the proposed method increases.

6 Conclusions

6.1 Related Work

The automatic discovery of useful knowledge pieces is a topic of growing interest in the expert systems engineering community [21], [22], [23]. Our work differs from those mentioned before in the proposal of a combined mechanism for rules obtaining, using self-organized maps based clustering and induction algorithms. On the other hand, the identification of the necessary processes to allow the autonomous assimilation of the knowledge pieces generated by the expert system. Knowledge discovery integration process models based on connectionist models [24], [25], [26], reasoning models based on cases [27], not expected patterns generation models [28], genetic algorithms [29] and technical categorization heuristics [30], have been proposed recently in order to dispose automatic processes for incremental improvement of the intelligent systems response applied to the specific problems resolution. This proposal differs from the ones mentioned above, in the fact that it proposes a knowledge discovery integration model (rules centered) with expert systems environment, identifying the technology needed to be used to solve this integration.

6.2 Future Research

In the different processes and how these processes interact with the different bases some problems have been identified in whose solution is foreseen to work: In the Inducer: how to use the support groups to provide a degree of credibility (trust) to the knowledge piece (rule) generated. In the Conceptual Labeler: [a] define the treatment to give to attributes values of concepts that are in the discovered rules but not in the Concepts Dictionary that emerges from the original Knowledge Base of the Knowledge Based System and [b] how to rewrite the ownership to a certain group (right part of the rule) in terms of values of attributes of well-known concepts when the knowledge pieces (rules) result from applying the Inducer to the Cluster. In the Knowledge Integrator it should be defined the treatment to apply when the integration process between the rules of the Knowledge Base and the discovered rules arise: [a] conditions of dead point, [b] recurrent rules, [c] redundant rules, [d] contradictory rules, and [e] rules with conflicts of support evidence, among others. "A priori" measures should be developed to establish the quality of the knowledge discovery process and the degree of integrability to the existent Knowledge Base.

References

1. Sierra, E., Hossian, A. y García-Martínez, R. 2003. *Sistemas Expertos que Recomiendan Estrategias de Instrucción. Un Modelo para su Desarrollo*. Revista Latinoamericana de Tecnología Educativa. 1(1): 19-30.
2. Sierra, E., García-Martínez, R., Hossian, A., Britos, P. y Balbuena, E. 2006. *Providing Intelligent User-Adapted Control Strategies in Building Environments*. Research in Computing Science Journal, 19: 235-241
3. Bermejo, F., Britos, P., Rossi, B y García Martínez, R. 2002. *Sistema de Asistencia para la Configuración de Ventiladores OAF en Neonatolgía*. Revista del Instituto Tecnológico de Buenos Aires. 28: 24-68.
4. Gómez, S., Perichinsky, G. y Garcia Martinez, R. 2001. *Un Sistema Experto Legal para la Individualización y Acuerdos para Penas*. Proc. Simposio Argentino de Informática y Derecho. pp. 23-33.
5. Diez, E., Britos, P., Rossi, By García-Martínez, R. 2003. *Generación Asistida del Mapa de Actividades de Proyectos de Desarrollo de Software*. Reportes Técnicos en Ingeniería del Software. 5(1):13-18.
6. García-Martínez, R. y Britos, P. 2004. *Ingeniería de Sistemas Expertos*. Editorial Nueva Librería.
7. Debenham, J. 1990. *Knowledge Systems Design*. Prentice Hall.
8. Debenham, J. 1998. *Knowledge Engineering: Unifying Knowledge Base and Database Design*. Springer-Verlag.
9. Michalski, R. Bratko, I. Kubat, M (eds.) 1998. *Machine Learning and Data Mining, Methods and Applications*, John Wiley & Sons Ltd, West Sussex, England

10. Grossman, R., Kasif, S., Moore, R., Rocke, D. and Ullman, J. 1999. *Data Mining Research: Opportunities and Challenges, A Report of three NSF Workshops on Mining Large, Massive, and Distributed Data*, January 1999, Chicago
11. Jensen D. 2002. *Knowledge Evaluation. Handbook of Data Mining and Knowledge Discovery*. Kloesgen, W. and J. Zytkow (Eds.). Oxford: Oxford University Press
12. Neville J. and D. Jensen 2002. Supporting relational knowledge discovery: Lessons in architecture and algorithm design. Proc. Data Mining Lessons Learned Workshop, 19th International Conference on Machine Learning.
13. Jensen D., Y. Dong, B. Lerner, E. McCall, L. Osterweil, S. Sutton Jr., and A. Wise 1999. *Coordinating agent activities in knowledge discovery processes*. In Proc. International Joint Conference on Work Activities Coordination and Collaboration. pp. 137-146.
14. Britos, P., Hossian, A., García Martínez, R. y Sierra, E. 2005. *Minería de Datos Basada en Sistemas Inteligentes*. 876 páginas. Editorial Nueva Librería. ISBN 987-1104-30-8.
15. García Martínez, R. y Borrajo Millán, D. 2000. *An Integrated Approach of Learning, Planning and Executing*. Journal of Intelligent and Robotic Systems 29(1):47-78.
16. Grosser, H., Britos, P. y García-Martínez, R. 2005. *Detecting Fraud in Mobile Telephony Using Neural Networks*. Lecture Notes in Artificial Intelligence 3533: 613-615.
17. Felgaer, P., Britos, P. and García-Martínez, R. 2006. *Prediction in Health Domain Using Bayesian Network Optimization Based on Induction Learning Techniques*. International Journal of Modern Physics C 17(3): 447-455.
18. Cogliati, M., Britos, P. y García-Martínez, R. (2006). *Patterns in Temporal Series of Meteorological Variables Using SOM & TDIDT*. Springer IFIP Series.
19. Rancán, C. 2004. *Arquitectura de Sistema Híbrido de Evaluación del Alistamiento de Unidades Navales Auxiliares*. Reportes Técnicos en Ingeniería del Software. 6(1): 45-54.
20. Pesado, P. Feierherd G. y Pasini, A. 2005. *Requirement Specifications for Electronic Voting System*. Journal of Computer Science & Technology, 5(4): 312-319 .
21. Hoffmann, F., Baesens, B., Mues, C. and Vanthienen, J. 2006. *Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms*. European Journal of Operational Research. (in press).
22. Cao, H., Recknagel, F. Joo, G., Kim, D. 2006. *Discovery of predictive rule sets for chlorophyll-a dynamics in the Nakdong River (Korea) by means of the hybrid evolutionary algorithm HEA*. Ecological Informatics, 1(1): 43-53.
23. Podgorelec, V., Kokol, P., Stiglic, M., Heričko, M., Rozman, I. 2005. *Knowledge discovery with classification rules in a cardiovascular dataset*. Computer Methods and Programs in Biomedicine, 80: S39-S49.
24. Huang, M., Tsou, Y., Lee, S. 2006. *Integrating fuzzy data mining and fuzzy artificial neural networks for discovering implicit knowledge*. Knowledge-Based Systems, 19(6): 396-403.
25. Kasabov, K. 2006. *Adaptation and interaction in dynamical systems: Modelling and rule discovery through evolving connectionist systems*. Applied Soft Computing, 6(3): 307-322.
26. Carpenter, G., Martens, S., Ogas, O. 2005. *Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks*. Neural Networks, 18(3): 287-295.
27. Liu, D., Ke, C. 2006. *Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning*. Expert Systems with Applications. (in press).
28. Moreno, M., Quintales, L., García, F., Polo, J. 2004. *Building knowledge discovery-driven models for decision support in project management*. Decision Support Systems, 38(2): 305-317.
29. Kim, M., Han, I. 2003. *The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms* Expert Systems with Applications, 25(4): 637-646.
30. Leigh, W., Modani, N., Purvis, R., Roberts, R. 2002. *Stock market trading rule discovery using technical charting heuristics*. Expert Systems with Applications, 23(2): 155-159.