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

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 vocabula. / 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,*

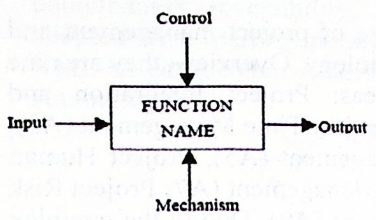
P = {P1, P2, ..., Pm} *is the finite set of places;*

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} U {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;*

$M0 : P \rightarrow \{0, 1, 2, 3, \ldots\}$ *is the initial marking.*



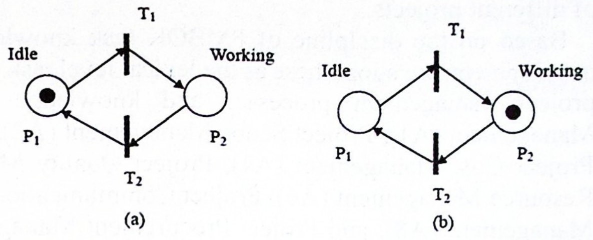


Fig.3 Component of the IDEF0 **Fig.4** Component of Petri net Model

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.

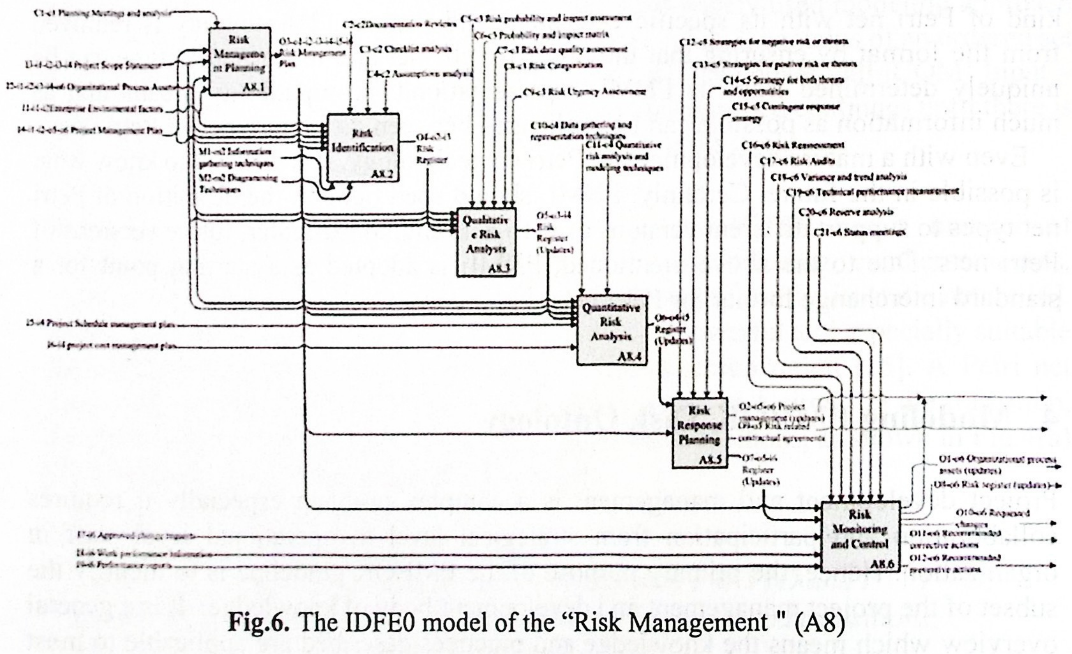
Even with a mature development on Petri net technology, it is difficult to know what is possible in the future. Certainly, PNML should shed light on the definition of Petri net types to support different versions of Petri nets and, in particular, future versions of Petri nets. Due to the above mentioned, 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.



Risk Management Planning
A8.1
Risk Identification
A8.2
Qualitative Risk Analysis
A8.3
Quantitative Risk Analysis
A8.4
Risk Response Planning
A8.5
Risk Monitoring and Control
A8.6
C1-c1 Planning Meetings and analysis
I1-I-I2-I3-I4 Project Scope Statement
I4-I5-I6 Project Management Plan
M1-m2 Information gathering techniques
M2-m2 Diagramming Techniques
C2-c2 Documentation Reviews
C3-c2 Checklist analysis
C4-c2 Assumptions analysis
C5-c3 Risk probability and impact assessment
C6-c3 Probability and impact matrix
C7-c3 Risk data quality assessment
C8-c3 Risk Categorization
C9-c3 Risk Urgency Assessment
C10-c4 Data gathering and representation techniques
C11-c4 Quantitative risk analysis and modeling techniques
C12-c5 Strategies for negative risk or threats
C13-c5 Strategies for positive risk or threats
C14-c5 Strategy for both threats and opportunities
C15-c5 Contingent response strategy
C16-c6 Risk Reassessment
C17-c6 Risk Audits
C18-c6 Variance and trend analysis
C19-c6 Technical performance measurement
C20-c6 Reserve analysis
C21-c6 Status meetings
O1-o2-o3 Risk Register
O5-o3-o4 Risk Register (Updates)
O6-o4-o5 Register (Updates)
O2-o5-o6 Project management plan (updates)
O7-o5-o6 Register (Updates)
O1-o6 Organizational process assets (updates)
O4-o6 Risk register (updates)
O10-o6 Requested changes
O11-o6 Recommended corrective actions
O12-o6 Recommended preventive actions
I5-i4 Project Schedule management plan
I6-i4 project cost management plan
I7-i6 Approved change requests
I8-i6 Work performance information
I9-i6 Performance reports

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

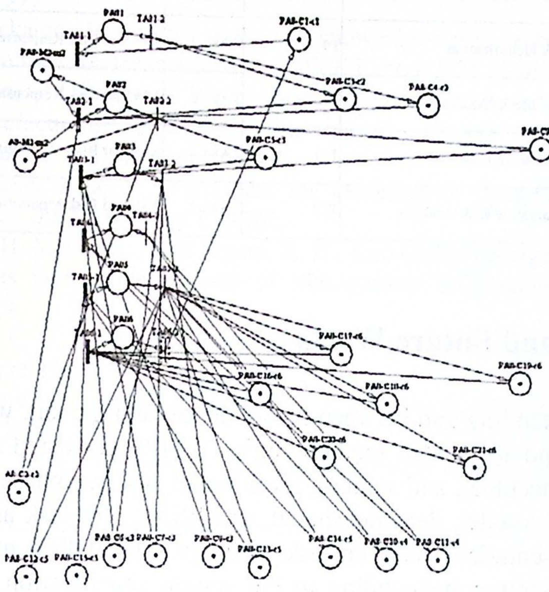


Fig.7. The PN model of the "Risk Management " (A8)

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.