東海大学大学院 平成26年度 博士論文

Knowledge Intensive Organization Model in Virtual Environment based on CommonKADS Methodology

指導　　　濱本　和彦　　　教授

東海大学大学院~総合理工学研究科

総合理工学専攻

BOONPRASERT SURAKRATANASAKUL

Abstract

This doctoral thesis presents a knowledge intensive organization model in virtual space based on CommonKADS methodology. One of challenges in knowledge engineering is analysis and organization of knowledge finding with an appropriate empirical methodology. Although some methodologies are powerful, many of them are passive, in the sense that only a few suggestions are made by the environment. From the survey, most of system lack of explain how an organization uses it knowledge is built up, collaborative work supportability and interface understandable manner. To compare with other techniques, it has complicated models and most of frameworks are non-standardization. In this thesis, I propose a novel knowledge intensive organization model in a virtual space based on knowledge model and organization model in CommonKADS framework, which by developing knowledge schema as a part of analysis process covering knowledge management level and knowledge object level. The benefit of the proposed method is useful and practical guidelines for knowledge intensive organization. It provides the methods to obtain a thorough understanding of the structures and processes used by knowledge workers. This method is not only to support knowledge methodology with its environment, but also encourage a groupware by participating in geographically-distributed development that contributes to knowledge exchange and sharing.

Chapter 1 explains research overview and knowledge management concept. Knowledge management consists of many mechanisms that encourage a system with SECI concept. To realize mechanisms, knowledge engineers need to explore the existing knowledge and recreate knowledge intensive task for solving encounter situation. The problem is lack of suggestion and environment-driven convergence in the real world for an intellective-insight. An involvement of knowledge development environment is related to the concept of computer-supported cooperative work assist communicating, collaborating, and coordinating activities. The critical success factor of knowledge-developing is an environmental supportability that encourages development process for creation and usage of existing knowledge. Finally is an overview of research methodology and positioning.

Chapter 2 contains background concepts, including CommonKADS framework, architectural views and UML extension, Google™ APIs, and Keyhole markup language (KML). CommonKADS methodology is a technique helps knowledge engineers to clarify the structure of a knowledge-intensive task and specification of knowledge data. Unfortunately, its framework has an ambiguous in model perspective and symbol-notation. To fix the problem, I apply an architectural model and UML extension mechanisms to modify the knowledge schema for identification. By the way, the Google™ APIs provide an interface to the provided services, and generate a virtual space by loading the necessary components onto display space. These open services APIs allow customization of the virtual space output, including ability to add application specific data on the space and integration to the third-party components. Keyhole markup language (KML) is a descriptive markup language based on the syntax and file format of XML. KML is used for describing and storing geographical information that is associated with two and three-dimensional coordinates system.

Chapter 3 elaborates knowledge systematic schemas, which is recreated from CommonKADS by using an architectural model. In this study, I develop three schemas on knowledge discipline name: knowledge landscape, knowledge atlas, and knowledge systematic schema. The knowledge landscape describes knowledge model in knowledge management level, on the other hand, knowledge atlas defines organization model in knowledge object level. I conclude both schemas in one schema: knowledge systematic schema. It was implemented three levels of architectural views: physical view, logical view, and functional view and used UML extension for describe model and elements.

Chapter 4 explains knowledge realization. I propose a general scene-graph to implement the knowledge systematic schema regimen for the virtual environment. About the element description, I explain via tag-based schema by separate geographical information in KML and knowledge informatics in XML with encapsulation. The system extracts the information using DOM-parser and manipulates knowledge informatics with AJAX implemented module. I demonstrate this study by

developing a prototype system using Google^{TM} Earth APIs environment as virtual environment.

Chapter 5 contains evaluation and discussion. For experiment the proposed schema, I provide three strategies evaluation: feature comparison, questionnaire user response, and experimental process task. In feature comparison, the results shows that the strength point of proposed system is interface-wise guidance in knowledge-developing supportability dimension and collaborative work in interoperability dimension. For the questionnaire user response, the experimental results show the proposed method satisfies on supportability, usability, and utility in knowledge-developing process. Additionally, its convergent design improves knowledge methodological suggestion for wider user with various experiences. About the process of experimental task experiment, the result can be shown that learning does not take too much time, so that can easily learn in proposed system. Moreover, user can learn by themselves without material suggestion and learning with experience from environment suggestion satisfies same as material guideline.

Chapter 6 is conclusions and future works. In this study, I present a knowledge intensive organization model in virtual environment based on CommonKADS methodology. I demonstrate the proposed approach by prototyping a system developed in Google^{TM} Earth APIs environment as virtual environment. Experimental results show that its convergent design improves knowledge methodological suggestion for wider user with various experiences and the proposed method satisfies on supportability, usability, and utility in knowledge-developing process. The new propose of this thesis are the three knowledge schemas: (1) knowledge landscape schema for knowledge concept in abstract space, (2) knowledge atlas schema for organization aspect in real world space, and (3) knowledge systematic schema for knowledge management system. Finally, I demonstrated the prototype application that developed with knowledge systematic schema in virtual environment. The results of experiment show that the proposed system improves knowledge methodology in various experience user levels for supportability, usability, and utility. Additionally, its convergent design improves knowledge methodological suggestion for wider user with various experiences. Based on this study, the proposed system can be further improved by including schema that provides more complicated knowledge system and strategies for complex explanation in virtual space. Furthermore, implementation in portable device may provide flexibility in access and collaboration at diverse location.

Contents

Contents (continued)

Contents (continued)

List of Figures

List of Figures (continued)

List of Tables

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Prof. Dr. Kazuhiko HAMAMOTO, whose encouragement, valuable guidance and being patient with supporting me to complete this thesis. I am also thankful to the examination committees for going through such a document and giving the valuable feedbacks, comments, suggestions and review this thesis: Prof. Dr. Kiyoshi NOSU, Prof. Dr. Hidekazu TSUJI, Prof. Dr. Akira TOMONO, and Prof. Dr. Hiroshi ISHII.

I am thankful to all members in HAMAMOTO laboratory for their warmest relationship and friendly.

I am grateful to JICA for the scholarship. I wish to express my warm and sincere thanks many JICA in charge: Ms. Maria Sutherland and Miss Tomomi Nishigaki, for their assisting through my stay in Japan.

Finally, I offer my regard and blessings to all of those who supported me in many ways during completion of this thesis and during stay in Japan.

Boonprasert Surakratanasakul

Tokai University

Chapter 1

Introduction

This chapter introduces an overview of research. It consists of four main parts. Firstly, section 1.1 describes about the concept of knowledge management that is main basic concept for developing the prototype system. Section 1.2 is literature review related with: knowledge quantization mechanism, ontology definition, computer-supported collaborative work concept, survey of existing systems for proposed system definition, and knowledge modeling techniques comparison. Section 1.3 describes research purpose: problems and opportunities, research goal, and challenge. Finally, section 1.4 is thesis organization and overview of research methodology.

1.1 Concept of Knowledge Management

Knowledge Management (KM) is an important mechanism for managing knowledge of organization that embedded in people, processes, and information generators. It views knowledge as a valuable asset for value-adding organization. Many organizations define their own KM strategies for explicating, developing, and distributing knowledge in order to stay competitive and be innovative on proactive approach. The well-known definition of knowledge management of Davenport and Prusak is defined [1]: "*Knowledge Management is the name given to a set of systematic and disciplined actions that an organization can take to obtain the greatest value from the knowledge over which it disposes.*"

In knowledge management, the words "Data", "Information", and "Knowledge" are three often-encountered words that belong closely together, seem to have slightly different meanings [2]:

Data are the uninterpreted signals that reach our senses every minute by the zillions. A red, green, or yellow light at an intersection is one example. Computers are full of data: signals consisting of strings of numbers, characters, and other symbols that are blindly and mechanically handled in large quantities.

Information is a data equipped with meaning. For a human car driver, a red traffic light is not just a signal of some colored object. It is interpreted as an indication to stop.

Knowledge is the whole body of data and information that people bring to bear to practical use in action, in order to carry out tasks and create new information. Knowledge adds two distinct aspects: First, a sense of <u>Purpose</u>, since knowledge is the "intellectual machinery" used to achieve a goal; Second, a <u>Generative Capability</u>, because one of the major functions of knowledge is to produce new information. It is not accidental, therefore that knowledge is proclaimed to be a new "factor of production"

	Characteristic	Example
Data	Uninterpreted Raw	…_ _ _...
Information	meaning attached to data	S O S
Knowledge	- attach purpose and competence to information - potential to generate action	Emergency -> start to rescue

Figure 1.1: Distinction between Data, Information, and Knowledge.

In information society, knowledge systems seem as an important mainstream technology that needs to convert the art and craft of knowledge systems building into a real scientific discipline. Organization and task analysis are knowledge-engineering activities that directly hook up with business administration and managerial aspects. A recent field that has emerged in business administration is knowledge management. It takes knowledge as a central subject for organizational decision making in its own right, and attempts to deal with the management control issues regarding leveraging knowledge.

1.1.1 Fundamental Process of Knowledge Management

Amrit defined the three fundamental processes of knowledge management as [3]:

- **Knowledge Acquisition**: The process of development and creation of insights, skills, and relationships. Knowledge formalization tools are examples of direct knowledge acquisition. Data capture tools with filtering abilities, intelligent databases, note-capture tools, and electronic whiteboards are examples of information technology components that can support indirectly knowledge data acquisition.

- **Knowledge Sharing**: Disseminating and making available what is already known. A decision support system that provides a novice physician best clinical practice is an example of knowledge that is being shared with that clinical agent. Possibly, the sharing process has to be optimized on the specific context.

- **Knowledge Utilization**: Learning is integrated into the organization. Whatever is broadly available throughout the company can be generalized and applied, at least in part, to new situations.

Figure 1.2 shows the fundamental process of knowledge management: knowledge acquisition, knowledge sharing, and knowledge utilization. Figure 1.3 shows the continuous process enabling users, who are utilizing the system for accomplishing their processes, to add, in the same time, new knowledge and share it.

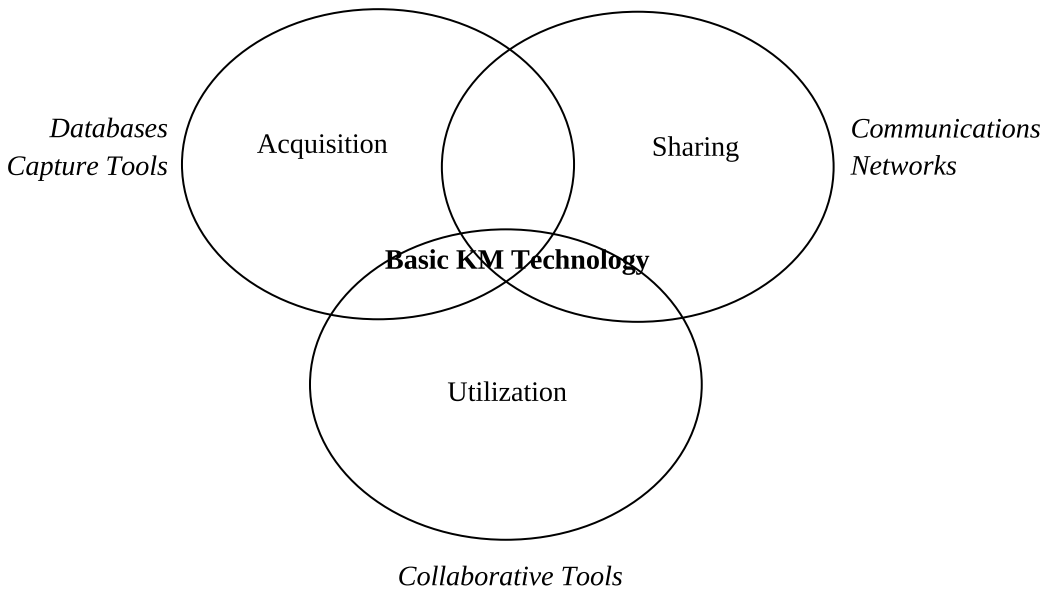


Figure 1.2: Basic of Knowledge Management Technology.

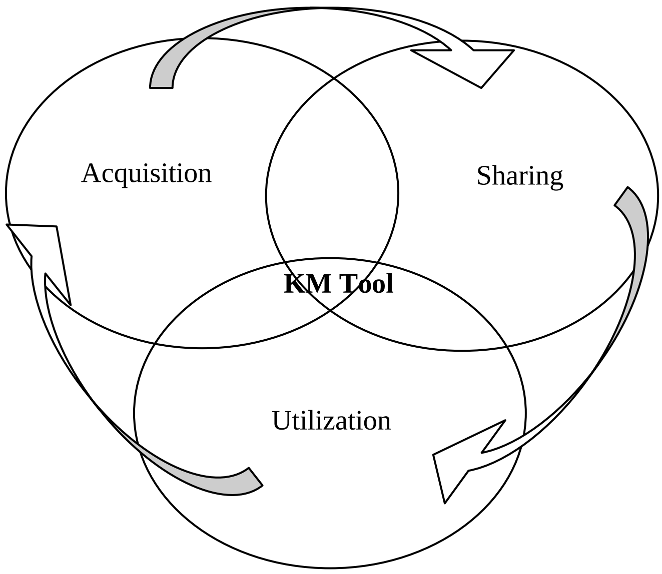


Figure 1.3: Knowledge Management Tool.

1.1.2 Knowledge Conversion Process

The classical view organization knowledge consists of explicit and tacit knowledge. Explicit knowledge can be expressed in words and numbers and shared in the form of data, scientific finding, product specifications, manuals, universal principles, and etc. This kind of knowledge can be readily transmitted across individuals formally and systematically. Tacit knowledge is highly personal and hard to formalize, making it difficult to communicate or shared with others. Subjective insights, intuitions and hunches fall into this category of knowledge. Furthermore, tacit knowledge is deeply rooted in an individual's actions and experience, as well as in the ideals, values or emotions.[4] Table 1.1 shows explicit knowledge versus tacit knowledge.

Table 1.1: Explicit Knowledge versus Tacit Knowledge.

Explicit Knowledge	Tacit Knowledge
• Knowing about (objective knowledge) • Rationalization of facts; formal methods. • Easy to codify and transfer. • Articulated knowledge (explicit knowledge assets) maybe moved instantaneously anytime anywhere by information technologies. • Codified knowledge maybe proactively disseminated to people who can use specific forms of knowledge. • Knowledge that has been made explicit can be discussed, debated, and improved. • Making knowledge explicit makes it possible to discover knowledge deficiencies in the organization.	• Knowing how (subjective knowledge) • Systems of ideas, perceptions, experience. • Difficult to transfer. • Relatively easy and inexpensive to begin. • Employees may respond well to recognition of the (claimed) knowledge. • Likely to create interest in further knowledge management processes.

In area of knowledge management, it has been pointed out based upon old work in philosophy, by the way a large part of knowledge is not explicit but tacit.[5] That is, knowledge is often not explicitly describable by the people who possess it, nor is it easy to explain and to formalize in books or manuals. Instead, it is a "background" capability, partly unconscious and stemming from experience that is used in problem-solving and other human task. "The Knowledge-Creating Company", Nonaka and Takeuchi [6] have built a whole theory about knowledge and its creation, on the basis of this distinction between tacit and explicit knowledge. As shown in Figure 1.4, four models of knowledge production are identified:

from \ to	*Tacit Knowledge*	*Explicit Knowledge*
Tacit Knowledge	**Socialization** Shares experience, discusses ideas, opinions	**Externalization** Articulate experience in formal model; embed experience into equipment software, etc.
Explicit Knowledge	**Internalization** Convert models and formulate into tacit skills, learn/teach how to use equipment	**Combination** Re-formulate formal models and data, converts codes, etc.

Figure 1.4: Knowledge Conversion Process.

1. from tacit to tacit knowledge (= **Socialization**): we can teach each other by showing rather than speaking about the subject matter;

2. from tacit to explicit knowledge (=**Externalization**): knowledge-intensive practices are clarified by putting them down on paper, formulate them in formal procedures, and the like;

3. from explicit to explicit knowledge (=**Combination**): creating knowledge through the integration of different pieces of explicit knowledge;

4. from explicit to tacit knowledge (=**Internalization**): performing a task frequently leads to a personal state where we can carry out a task successfully without thinking about it.

Organizational knowledge creation continuously needs all four types of knowledge production. The aim of knowledge management is to properly facilitate and stimulate these knowledge processes, so that an upward, dynamic spiral of knowledge emerges. This is a unique feature of knowledge engineering, because there is hardly any other mature scientific methodology capable of externalization tacit knowledge. Also the combination of knowledge is well supported in knowledge engineering, e.g., through libraries of reusable task and domain models. The importance of tacit knowledge is nowadays widely acknowledged in knowledge engineering and management.

1.1.3 Lifecycle of Knowledge

There are many frameworks for knowledge management. Most of all have in common their intension to cover the complete lifecycle of knowledge within the organization.[7] Typically, the following activities with respect to knowledge and its management are distinguished by and depended on many authors. Figure 1.5 shows the activities in knowledge management and the associated knowledge-value chain.

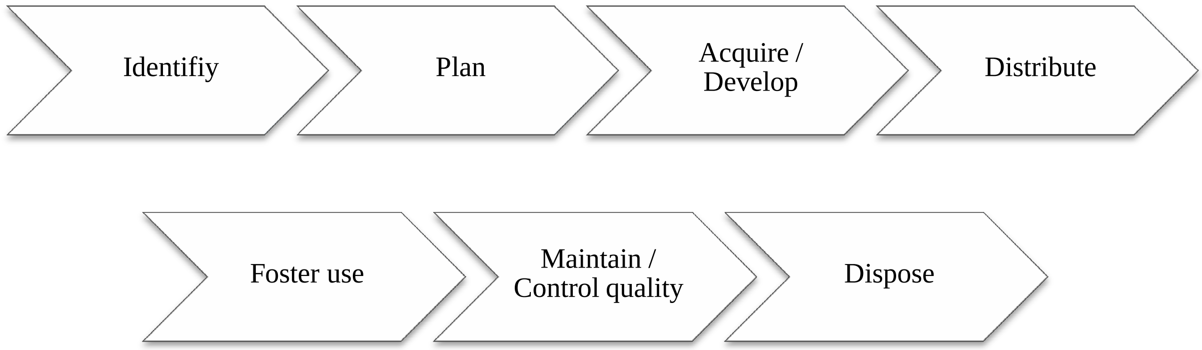


Figure 1.5: Activities in KM and the Associated Knowledge-value Chain.

- **Identify**: internally and externally existing knowledge.
- **Plan**: what knowledge will be needed in the future.
- **Acquire and/or develop**: the needed knowledge.
- **Distribute**: the knowledge to where it is needed.
- **Foster the application**: of knowledge in the business processes of the organization.
- **Control**: the quality of knowledge and maintain it.
- **Dispose**: of knowledge when it is no longer needed.

The practical definition of knowledge management is: a framework and tool set for improving the organization's knowledge infrastructure, aimed at getting the right knowledge to the right people in the right form at the right time.[8]

Obviously, knowledge management is not a one shot activity. It is embedded in a cycle model of the Learning Organization. This is based, for example, on Argyris model of "double loop" organizational learning. The first loop is direct learning about an application, product, or activity. The second loop runs on top of that and is learning about knowledge and learning itself whereby the mission, goals, and strategy of the organization act as the driving force. Knowledge management helps the organization to obtain feedback and continuously learn from its own experiences, on the basis of which its knowledge infrastructure for the future.

1.2 Literature Review

The related literature review consists of: knowledge quantization mechanism, ontology definition, computer-supported collaborative work concept, survey of existing systems for proposed system definition, and knowledge modeling techniques comparison.

1.2.1 Knowledge Quantization Mechanism

According to Webster's dictionary defined, "Knowledge is the fact or condition of knowing something with familiarity gained through experience or association; acquaintance with or understanding of a science, art, or techniques; the sum of what is known: the body of truth, information, and principles acquired by mankind." [9] In philosophy, cognition belongs to consciousness category. Danah Zohar [10] thinks the consciousness moves under the quantum mechanics rule, and his quantum management science has caused response and discussion in the international management domain in recent years. According to the understanding about the light in physics, Verna Allee [11] proposed that the knowledge has "the wave-particle duality". We may understand the knowledge for the entity and the process from different angles. The particle identifies the body knowledge, while the wave refers to the process knowledge and terms of tacit knowledge. So the knowledge is one kind of field material which has quantum attributes and characteristic.

Quantum management thinking changes the level and border of knowledge energy: According to the DIKW hierarchy model (The Data Information Knowledge and Wisdom Hierarchy) about human mind proposed by Russell Ackoff [12] used figure 1.6 shows the transition from data to information to knowledge and to wisdom. They thought understanding could facilitate the transition of process from one to another. The transition from data to wisdom is often along with the transformation from explicit knowledge to tacit knowledge through learning.

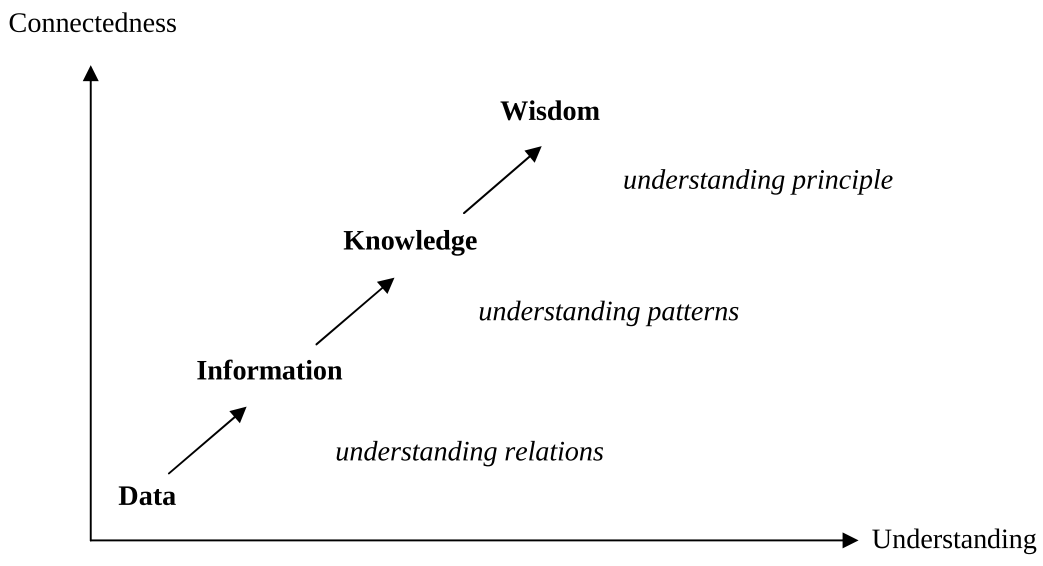


Figure 1.6: Transition from Data to Information to Knowledge and to Wisdom. [13]

Conceptually, the framework of knowledge quantization is generalized to a quantization spiral that comprises (1) quantization, (2) construction, (3) tailoring, and (4) re-quantization.[14] Figure 1.7 shows the spiral of knowledge quantization mechanism.

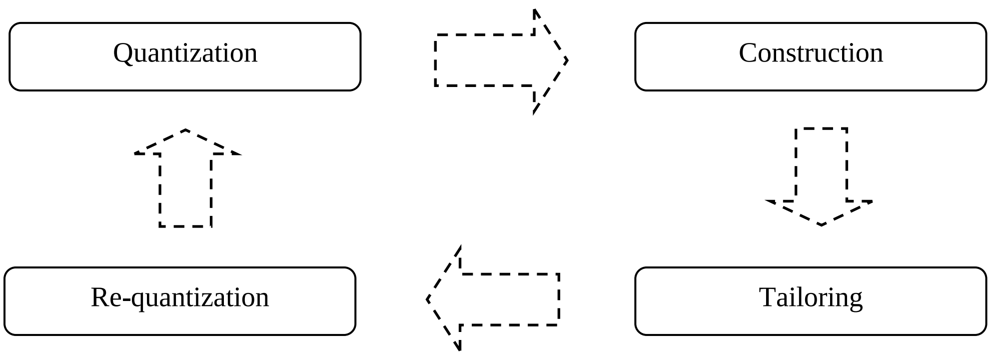


Figure 1.7: Knowledge Quantization Mechanism.

Figure 1.8 expresses the knowledge quantization mechanism in detail and compare activities on the knowledge abstract and real world knowledge.

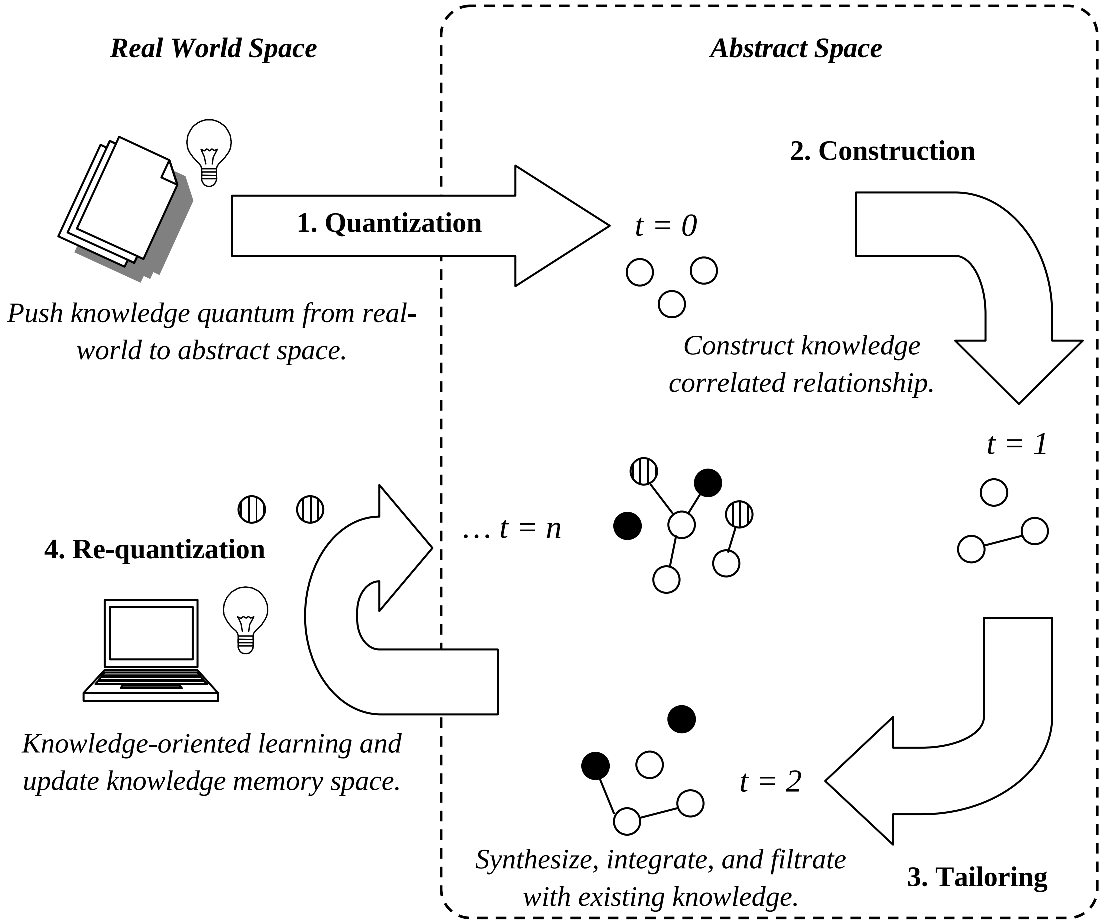


Figure 1.8: Knowledge Quantization Mechanism Spiral. [15]

1.2.2 Ontologies

Ontologies are put forward as a means to share knowledge bases between various knowledge based systems. The main motivation behind ontologies is that allows for sharing and reuse of bodies of knowledge in a computational form, and develop a library of reusable ontologies in a standard formalism that each system developer was supposed to adopt.[16] The essence of ontology is based on the related definition: ontology is a formal, explicit specification of a shared conceptualization. “Conceptualization” refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena. “Explicit” means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domain, the concepts are diseases and symptoms, the relations between them are casual and a constraint is that a disease cannot cause itself. “Formal” refers to the fact that the ontology should be machine readable, which excludes natural language. “Shared” reflects the notion that ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.[17-19]

Ordinally, the term ontology comes from philosophy where it is employed to describe the existence of begins in the world. Artificial Intelligence (AI) deals with reasoning about models of the world. Therefore, it is not strange that AI researchers adopted the term ontology to describe what can be (computationally) represented about the world in a program.[20]

Ontologies are also essential to the development and use of intelligent systems, particularly for the interoperation of heterogeneous systems. They are responsible for informing about the domain vocabulary and explaining the meaning that interacting systems attribute to terms. Furthermore, they facilities the domain model construction since it is through the ontology that the vocabulary of terms and relations, with which it is possible to model the domain, is provided.[21]

1.2.3 Computer-Supported Cooperative Work

Computer-supported cooperative work (CSCW) is an idea that people should be able to work together in a group, but do not have to be at one pace or time. “CSCW looks at how groups work and seeks to discover how technology (especially computers) can help them work”. Because people should interact with each other by some means of a communication model, which is the same for all attendees, software packages have been constructed to help the interaction between the users. The programs and additional hardware are called “Groupware”.[22-23]

Groupware is a computer-based system that supports groups of people engaged in a common task or goal, and provides an interface to share the environment. Collaboration technologies, cooperative systems, coordination tools, group support systems, etc., are synonymous with the groupware technology. By mediating human interaction and communication processes, groupware systems have the potential to bring about the dramatic changes to the social functioning of individuals, groups, and organizations. The goal of groupware is to assist groups in communicating, in collaborating, and in coordinating their activities. Because of ontology building by a group of people geographically separated over the world is likely to increase in the future; CSCW aspects by Groupware are becoming more relevant for the ontology development tools. [24-26]

1.2.4 Survey of Existing Systems and Proposed System Definition

This section presents a survey of existing system in current market comparison with the proposed system. In this study, I chose Protégé-2000 and WebODE for comparison. A survey is divided into following clusters:

- **Development tool feature**: includes tools, environments and suites that can be used for building a new from scratch or reusing existing knowledge. Apart from the common edition and browsing functionality, included documentation, exportation and importation from different formats, graphical view, and libraries.

- **Merge and integration feature**: appeared to solve the problem of merging or integrating different on the same domain. This need appears when two companies or organizations are merged together, or when it is necessary to obtain a better quality schema from other existing in the same domain.

- **Evaluation feature**: appear as support tools that ensure schema and its related technologies have a given level of quality.

- **Annotation feature**: the tool has been designed to allow users inserting and maintaining (semi)automatically schema.

- **Storage and querying feature**: the tool has been created to allow using and querying easily. Use of the web as a platform for communicating knowledge have appeared in this context.

Figure 1.9 shows a comparison survey of three systems: the proposed system, Protégé 2000, and WebODE. A chart was arranged by number of features and system environments separated in each cluster. All survey features = 28 features: development environment 10 features, merge and integration 7 features, evaluation 4 features, annotation 4 features, and storage and querying 3 features. For more detail of survey are elaborated in an appendix.

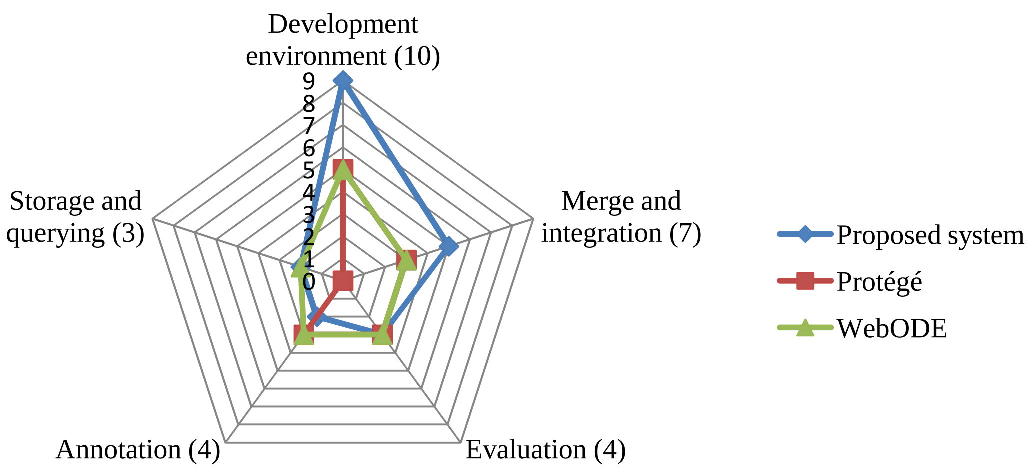


Figure 1.9: Survey of the Proposed System Comparison with Protégé 2000 and WebODE.

1.2.5 Knowledge Modeling Techniques Comparison

From knowledge modeling techniques were reviewed (CommonKADS, Protégé 2000, Multi-perspective, and UML), CommonKADS is the only techniques that can be considered a knowledge engineering methodology. All this technique supports object-oriented approach in modeling activities and their models are platform independent. CommonKADS, multi-perspective modeling and UML are considered as hybrid approach in modeling as opposed to Protégé which is not modeling tools in sense that use it to draw visual models or diagrams, but it is a tool that allows us to input the knowledge into its knowledge base. The modeling part of Protégé is already incorporate into the editing tool that could not be seen by the users. UML is a standard for modeling defined by OMG; where else the other techniques are not standardizing in a formal manner. All these techniques are fully documented in various forms. CommonKADS and UML is fully documented in books and reports, Protégé documentations are online at their website, multi-perspective modeling are documented by the respective modeling techniques. Most of these techniques are evolving; Protégé is undergoing further enhancement by the Protégé developers, multi-perspective by the respective technique developer and UML by the OMG members. These techniques are useful to model domains, ranging from medical, legal, engineering, business and up to social sciences. Protégé 2000 modeling technique supports Open Knowledge Base Connectivity (OKBC) knowledge model and can be adapted for editing models in different Semantic Web languages and supports RDF (Resource Description Framework) format for saving files. The modeling techniques and their features are listed in Table 1.2.

Table 1.2: Comparison of Knowledge Modeling Techniques.

Technique Feature	**CommonKADS**	**Protégé 2000**	**Multi-perspective**	**UML**
K.E. methodology	✓			
Object-oriented Approach	✓	✓	✓	✓
Platform Independent	✓	✓	✓	✓
Hybrid Approach	✓		✓	✓
Editor Tool		✓		
Standard Modeling Language				✓
Documentation	✓	✓	✓	✓
Evolving		✓	✓	✓
Domain	Medical, legal, engineering, business and up to social sciences	Medical, legal, engineering, business and up to social sciences	Medical, legal, engineering, business and up to social sciences	Medical, legal, engineering, business and up to social sciences
Other features (OKBC, RDF, Semantic web)		✓		

1.3 Research Purpose

The research propose starts from the source of problems and opportunities from literature review, survey, and comparison, then identify the research achieve goals with the features of acquired system, and develop through the research challenge.

1.3.1 Problems and Opportunities

- **[P-1]** Organization knowledge is a key asset in an organization but it is often tacit and private. From the survey, most systems lack of explain how an organization uses it knowledge is built up.
- **[P-2]** From the survey, many systems lack of an interface understandable manner and suggested usability in user perform.
- **[P-3]** From the survey, some system has not methodological support and lack of collaborative work to improve knowledge exchange.
- **[P-4]** From the comparison, some technique has complicated models and most of frameworks are non-standardization language for knowledge-developing process.

1.3.2 Research Goals

Overview purposes of the system are identified as followed:

- **[G-1]** Aim to develop useful and practical guidelines for knowledge intensive organization by develop the schema is to get acquainted with the system and to assess the amount of foreknowledge needed.
- **[G-2]** Enables one to spot the opportunities and bottlenecks in how organizations develop, distribute and apply their knowledge resources, and so gives tools for corporate knowledge management.
- **[G-3]** Provide the methods to obtain a thorough understanding of the structures and processes used by knowledge workers even where much of their knowledge is tacit leading to a better integration of information technology in support of knowledge work.
- **[G-4]** Designed concern how difficult is it to learn to work with the system and about the amount of knowledge required of the underlying knowledge representation language.
- **[G-5]** Build better knowledge system that easier to use, has a well-structured architecture, and simpler to maintain.

1.3.3 Research Challenges

- Finds an appropriate schematic for developing Knowledge Management System (KMS).
- Discover two dimensional knowledge spaces: abstract knowledge and real world knowledge, on hypothesis this connection can encourage the intellective insight.
- Supports and encourages knowledge management mechanisms.
- Improves communication and collaboration.
- Easy to use in non-experience users and beginner users, through the concept "everyone can use and share knowledge".
- Based on the fundamental process of knowledge management, knowledge conversion process, lifecycle of knowledge, and learning organization.

1.4 Thesis Organization

The thesis is organized as follow: Chapter 2 reviews about the background concepts: 2.1: CommonKADS framework, 2.2: architectural views and UML extension mechanisms, 2.3: the Google^{TM} API, and 2.4: Keyhole Markup Language (KML). Chapter 3 describes about knowledge schema covers both of concept and context levels. Chapter 4 describes knowledge realization, virtual space design, realized mechanisms, and system definition. Chapter 5 is evaluation and discussion. Finally, chapter 6 is conclusions and the future works.

1.4.1 Methodology

The flow chart diagram below in figure 1.9 shows the steps of the research methodology.

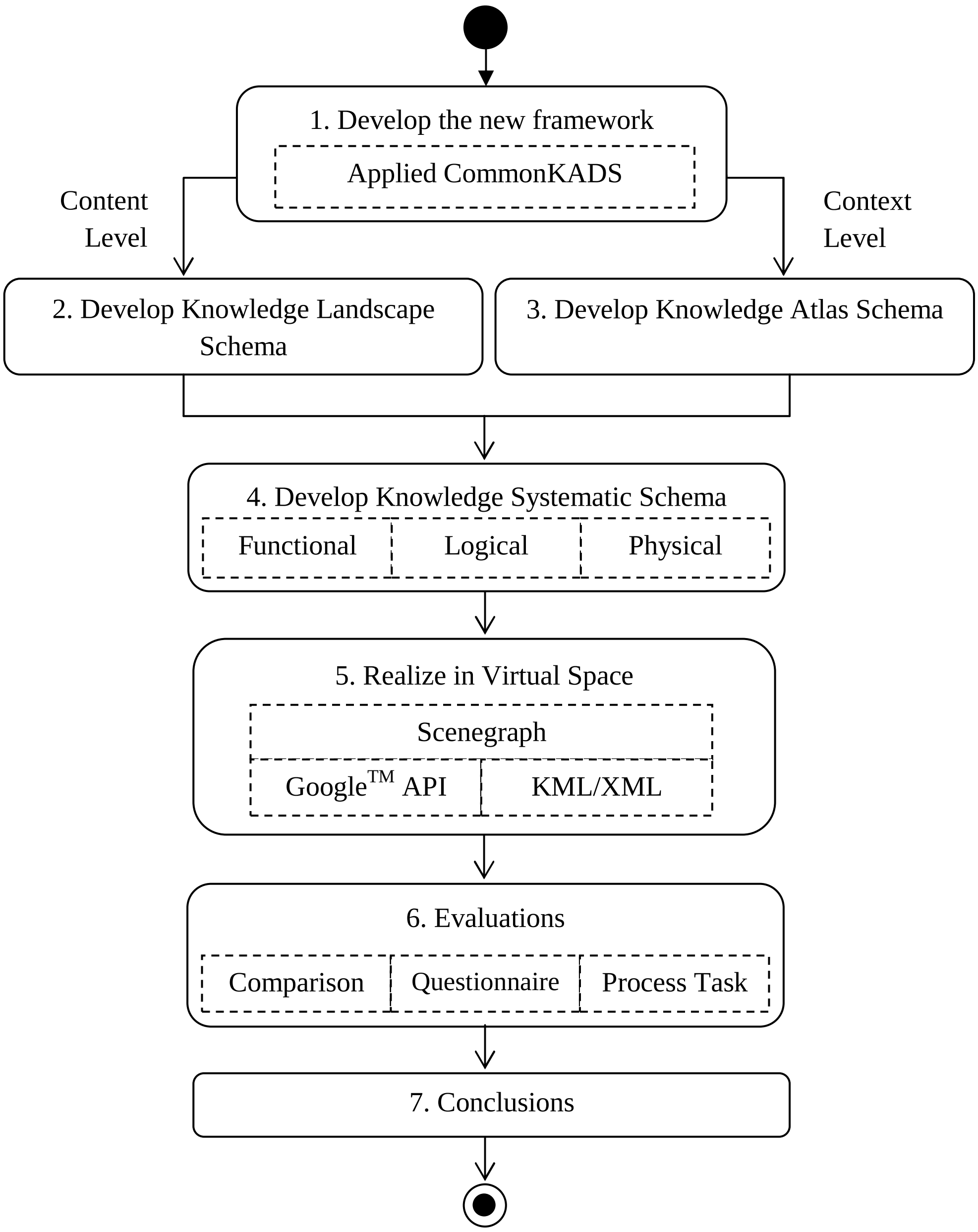


Figure 1.10: Research Methodology Overview.

Bibliography

[1] Davenport T.H., Prusak L., "*Working Knowledge: How Organizations Manage What They Know*", Harvard Business School Press, Boston, (1997).

[2] Wiig M., "*Knowledge Management – An Emerging Discipline Rooted in a Long History*", Journal of the American Society for Information Science, vol.53, no.12, pp.1009-1018 (1999).

[3] Amrit Tiwana, "*The Knowledge Management Toolkit*", Prentice Hall, (2002)

[4] Nonaka I., "*A Dynamic Theory of Organizational Knowledge Creation*", Organization Science, vol.5(1), pp.14-37 (1994).

[5] Herschel, R.T., Nemati, H. and Steiger, D., "*Tacit to Explicit Knowledge Conversion: Knowledge Exchange Protocols*", Journal of Knowledge Management, vol.35, pp.113-125.

[6] Nonaka I. and Takeuchi H., "*The Knowledge Creating Company*", Oxford: Oxford University Press (1995).

[7] R.Morkvenas, J.Bivainis, and M.Samoska, "*Analysis of Organization Knowledge Potential Content*", proceedings of the ITI2009 31th Int.Conf on Information Technology Interfaces, June 22-25, Cavtat, Croatia, pp.499-503 (2009).

[8] Jim Greer, Gordon McCalla, John Cooke, Jason Collins, Vive Kumar, Amdrew Bishop and Julita Vassileva, "*The Intelligent Helpdesk: Supporting Peer-Help in a University Course*", ITS'98, LNCS 1452, pp.494-503 (1998).

[9] A. Th. Schreiber, J.M. Akkermans, A. A. Anjewierden, R. de Hoog, N. R. Shadbolt, W. Van de Velde, and B. J. Wielinga, "*Knowledge Engineering and Management: The CommonKADS Methodology*", MIT Press, Cambridge, MA, (1999).

[10] Leif E. and Malone M., "*Intellectual Capital: Realizing Your Company's True Value by Finding Its Hidden Brainpower*", New York: Harper Business; pp.10-15 (1997).

[11] Bally J.M., "*Designing Workscape: an Inter-disciplinary Experience*", SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence, pp.10-15 (1994).

[12] Kingston, J. and Macintosh, A., "*Knowledge Management through Multi-Perspective Modeling : Representing and Distributing Organizational Memory*", Knowledge-Based Systems, vol.13, pp.121-131.

[13] Ackoff, R. L., "*From Data to Wisdom*", Journal of Applied Systems Analysis, vol. 16, pp.3-9 (1989).

[14] Toyoaki Nishida, "*Conversational Informatic An Engineering Approach*", John Wiley & Son Ltd, England (2007).

[15] Kubota H., Sumi Y., and Nishida T., "*Sustainable Knowledge Globe: a System for Supporting Content-oriented Conversation*", AISB, pp.80-86 (2005).

[16] Minoh M., and Nishiguchi S., "*Environment Media – In the Case of Lecture Archiving System*", KES 2003, vol. II, pp.1070-1076 (2003).

[17] Hidekazu Kubota, Satoshi Nomura, Yasuyuki Sumi, and Toyoaki Nishida, "*Sustainable Memory System Using Global and Conical Spaces*", Journal of Universal Computer Science, vol. 13, no. 2, pp.135-148 (2007).

[18] N. Fridman Noy, R. W. Fergerson, and M. A. Musen, "*The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility*", SMI Technical Report, Standford University, School of Medicine (2000).

[19] Fensel D., "*Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*", Springer (2001).

[20] P. Speel, N. R. Shadbolt, W. de Vries, P. van Dam,and K. O'Hara, "*Knowledge Mapping for Industrial Purposes*", In Proc Twelfth on Knowledge Acquisition, Modeling Management (KAW'99) (1999).

[21] A. Anjewierden, J. Wielemaker and C. Toussaint, Shelley, "*Computer Aided Knowledge Engineering*", Knowledge Acquisition, vol.4(1) (1992).

[22] Retana S.S. Guizzardi, Lora Aroyo, and Gred Wagner, "*Agent-Oriented Knowledge Management in Learning Environments: A Peer-to-Peer Helpdesk Case Study*", AMKM 2003, LNAI 2926, pp.57-72 (2003).

[23] Robert Woitsch and Dimitris Karagiannis "*Process-Oriented Knowledge Management Systems Based on KM-Services: The PROMOTE Approach*", PAKM 2002, LNAI 2569, pp.398-412 (2002).

[24] A. Th. Schreiber, B. J. Wielinga, R. de Hoog, J.M. Akkermans, and W. Van de Velde, "*CommonKADS: A Comprehensive Methodology for KBS Development*", IEEE Expert, 9(6):28-37, December (1994).

[25] Wilfried Post, Bob Wielinga, Robert de Hoog, and Guus Schreiber, "*Organizational Modeling in CommonKADS: The Emergency Medical Service*", IEEE Expert: Intelligent Systems and Their Applicationd, vol.12, November, pp.46–55 (1997).

[26] Cuena, J. and M. Molina, "*The Role of Knowledge Modeling Techniques in Software Development: a General Approach Based on a Knowledge Management Tool*", International Journal of Human-Computer Studies, vol.52, pp.385-421 (2000).

Chapter 2

Background Concepts

This chapter describes about research background concepts. It consists of four main parts. Firstly, section 2.1 describes about the concept of CommonKADS framework that applied for this research. Section 2.2 describes architectural model views and UML extension mechanisms for develop knowledge schemas. Section 2.3 describes the Google™ APIs and its features. Finally, section 2.4 describes about Keyhole markup language (KML) and their characteristic.

2.1 CommonKADS Framework

2.1.1 Knowledge Management with CommonKADS

The basic of knowledge management with CommonKADS approach is distinguish a management level and knowledge object level. For the knowledge management level, we see knowledge as a resource that has to manage just as any other resource. Knowledge-management level comprises management tasks.[1] Basically, this means that the resource has to be made available: - at the right time; at the right place; in the right shape; with the need quality; against the lowest possible cost.

To make knowledge management a viable enterprise, more flesh must be added to the skeletal model in figure 2.1. This means describing a process model for the management level and an object model for the object level.[2]

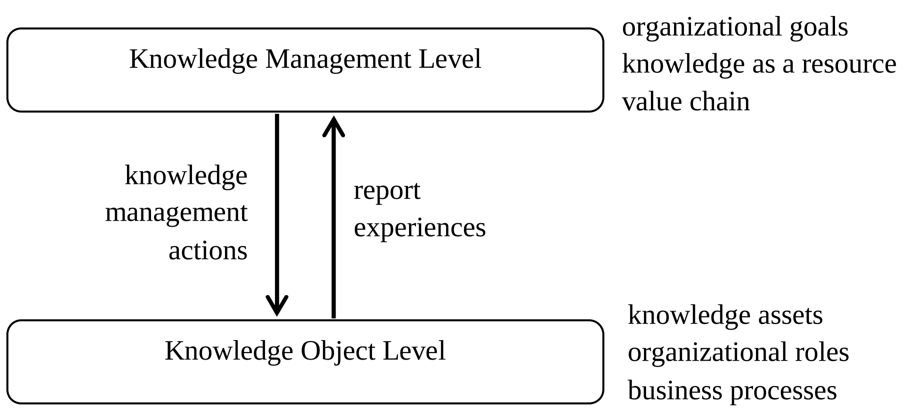


Figure 2.1: KM as Meta-level Activity acts on Object-level.

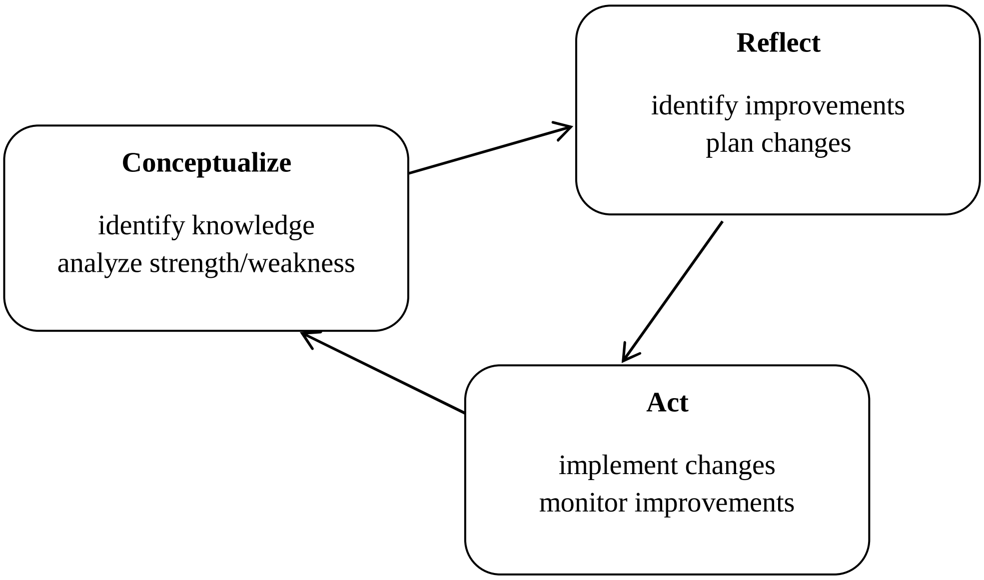


Figure 2.2: Cyclic Execution of three main KM activities: Conceptualize, Reflect, and Act.

Figure 2.2 shows the knowledge management is a cyclic process that consisting of three different types of management activities: conceptualize, reflect, and act.[3]

2.1.2 CommonKADS Principle

The CommonKADS enterprise originates from the need to build industry-quality knowledge systems on a large scale, in a structured, controllable, and repeatable way. When the CommonKADS work started back in 1983, there was little interest in such methodological issues. At that time, the prevailing paradigm for knowledge systems was rapid prototyping of one-shot applications, using special purpose hardware and software such as LISP machines, expert system shells, and so on.[4-5]

A methodology such as CommonKADS or any other software-development approach consists of a number of elements. These elements can be depicted graphically in the form of a pyramid see figure 2.3.[6] The methodological pyramid has five layers, where each consecutive layer: the “worldview” of the methodology. There are in fact the advertising slogans of an approach. These slogans need to be grounded in theory, methods tools and practical case studies which constitute the other four layers.[7] The slogans of CommonKADS can be formulated as a number of principles that form the baseline and rationale of the approach. The principles are based on the lessons learned about the knowledge-system development in the past.[8-9]

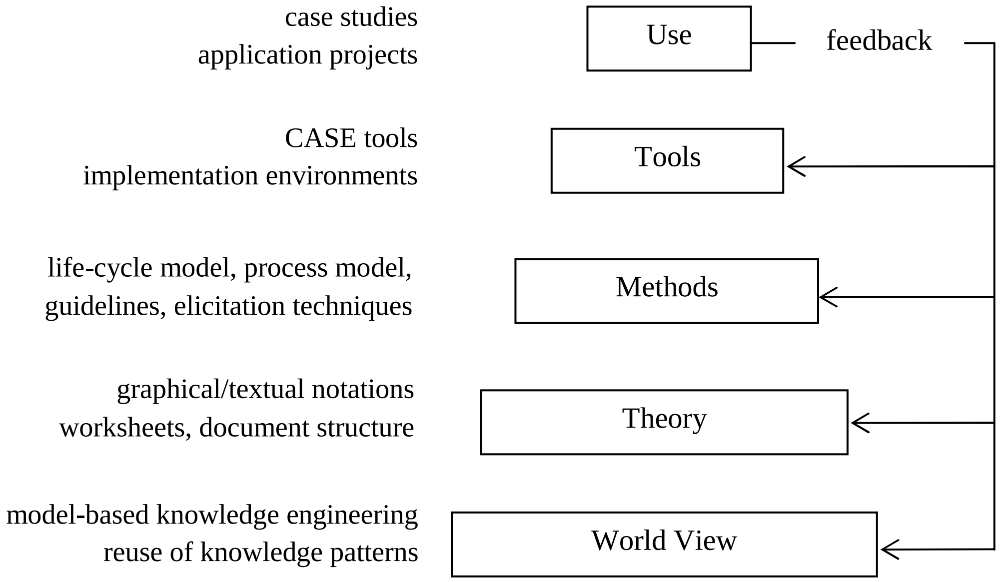


Figure 2.3: Building Blocks of CommonKADS Methodology.

CommonKADS model suite is the practical expression of the principle underlying knowledge analysis. It constitutes core of the CommonKADS knowledge-engineering methodology. Figure 2.4 shows three groups of levels. In each level has models for develop the own goal for target knowledge system. In context level, it consists of organization model, task model, and agent model. Concept level, it has knowledge and communication model. Artifact level has only one model: design model.[10-12]

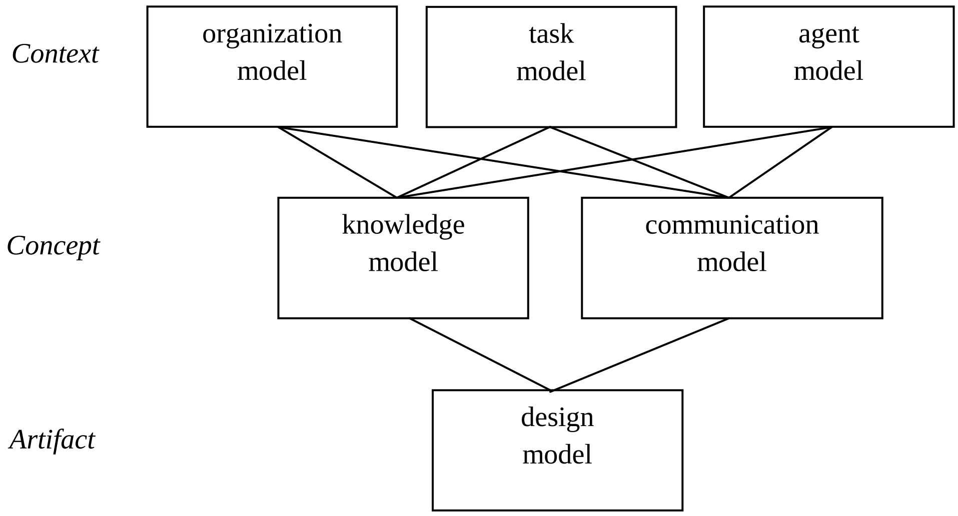


Figure 2.4: CommonKADS Model Suites.

2.1.3 Purpose of Models in CommonKADS

Table 2.1: Description of CommonKADS Model Levels.

Level	Description
Context	Analyze an organizational environment and the corresponding critical success factors for a knowledge system.
Concept	Yield the conceptual description of problem-solving functions and data that are to be handled and delivered by a knowledge system.
Artifact	Converts into a technical specification that the basis for software system implementation.

In practice, not always do all models have to be constructed. It depends on goals of project as well as experiences gained in running project.

Table 2.2: Description of CommonKADS Models.

Model	Description
	Context Level
Organizational model	Support the analysis of major features an organization, in order to discover problems and opportunities for knowledge system, establish their feasibility, and assess the impacts on the organization of intended knowledge actions.
Task model	Tasks are the relevant subparts of a business process. The task model analyzes the global task layout, its inputs and outputs, preconditions and performance criteria, as well as needed resources and competences.
Agent model	Agents are executors of task. An agent can be human, an information system, or any other entity capable of carrying out a task. The agent model describes the characteristics of agents, in particular their competences, authority to act, and constraints in this respect. Furthermore, it lists the communication links between agents in carrying out a task.

Table 2.2: Description of CommonKADS Models. (continued)

Model	Description
	Concept Level
Knowledge model	The purpose of knowledge model is to explicate in detail types and structures of the knowledge used in performing a task. It provides an implementation-independent description of role that different knowledge components play in problem-solving, in a way that is understandable for humans. This makes the knowledge model an important vehicle for communication with experts and users about the problem-solving aspects of a knowledge system, during both development and system execution.
Communication model	Since several agents may be involved in a task, it is important to model the communicative transactions between the agents involved. This is done by communication model, in a conceptual and implementation-independent way, just as with the knowledge model.
	Artifact Level
Design model	The above CommonKADS models together can be seen as constituting the requirements specification for the knowledge system, broken down in different aspects. Based on these requirements, the design model gives the technical system specification in terms of architecture, implementation platform, software modules, representational constructs, and computational mechanisms needed to implement the functions laid down in the knowledge and communication model.

2.2 Architectural Model Views and UML Extension

2.2.1 Architectural Model Views

Architectural model deals with the analysis and design of the high-level structure of system. It is the result of assembling a certain number of architectural elements in some well-chosen forms to satisfy the major functionality and performance requirements of system, as well as some other non-functional requirements such as reliability, scalability, portability, and availability. Architectural model deals with abstraction, with decomposition and composition, with style, and esthetics. To describe architecture, I use a model composed of multiple-view perspectives. In research, I propose two architectural model's views for develop knowledge modeling: Logical view and Functional view.[23]

Logical view primarily supports what the system should provide in terms of services. The system is decomposed into a set of key abstractions, taken mostly from the domain in the form of objects or object classes. It exploits the principles of abstraction, encapsulation, and inheritance.

Functional view supports how the system's elements work together seamlessly by usage of scenarios. For which, it describes the corresponding script like the inference structure and scenario in task knowledge of CommonKADS methodology in research.

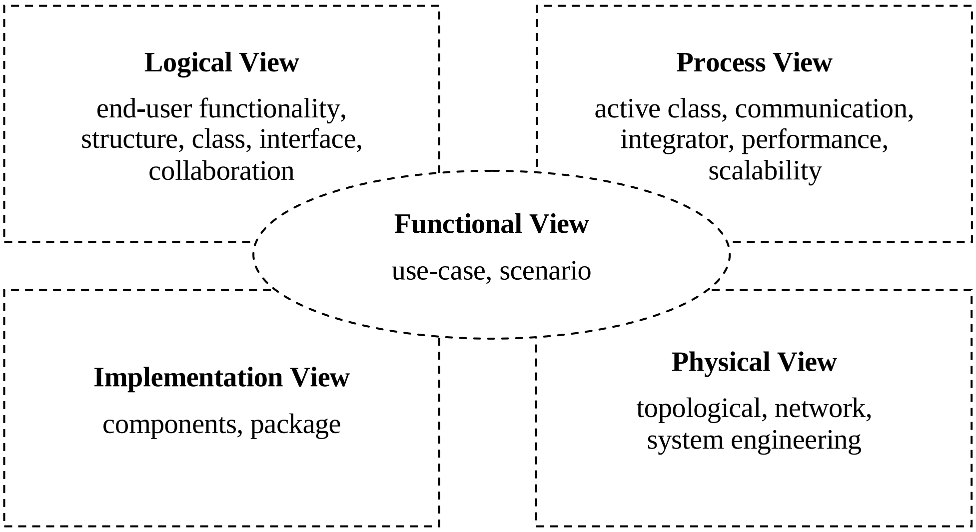


Figure 2.5: Architectural Model Views.

2.2.2 UML Extension Mechanisms

The extension mechanisms are provided by the UML in order to allow users to customize and extend the language to suite their particular needs. The standard extension mechanisms allow developer to adapt UML to accommodate new concepts are: Stereotype, Tagged Value, and Constraint.[24-25].

Stereotype is an extension of vocabulary of UML, which allows designer to create new building blocks from existing ones but specific to domain problem. Basically, all UML elements can be customized and/or extended by defining and naming using the stereotypes. General form of stereotypes is <<stereotype-name>>.

Tagged value is an extension of properties of a UML element which allows designer to create new information in that element's specification. It need not be always visible and can be contained e.g. in a database record associated to the object, which is not graphically represented in a diagram. General form is: {tag=value}.

Constraint is an extension of semantics of a UML element. It represents rules that apply to UML models. It may apply to one or more elements within the model. Designer may employ both predefined and user-defined constraints. Constraints may also be defined using the Object Constraint Language (OCL).

Nowadays the UML extension mechanisms are applied in several model systems for example; UML for Enterprise Application Integration (EAI), UML for CORBA, and UML for Enterprise Distributed Object Computing (EDOC), and etc.[26]

2.3 The Google™ APIs

The Google™ APIs reference includes a description of various interface, members, and google.earth functions in the Earth API. The google.earth namespace contains global functions that aid in process of using the Earth API interfaces. For examples, instantiation of Google™ Earth browser plug-in objects is done via the google.earth.createInstance method, event handling can be accomplished via the google.earth.addEventListener and google.earth.removeEventListener methods, etc.[27]

For the browser plugin-specific interface, interfaces whose names begin with GE allow for programmatic access to core plugin functionality and other miscellaneous options. For examples, GEGlobe, GENavigationControl, GEControl, GEOptions, GETime, GEEventEmitter, GETimeControl, GEPlugin, GETourPlayer, GEFeatureContainer, GESchemaObjectContainer, GEView, GEGeometryContainer, GEStyleSelectorContainer, GEWindow, etc.

For KML-based interfaces, interfaces whose name begins Kml- represent KML –related objects such as <Placemark> and <LookAt>. For examples, KmlAbstractView, KmlFolder, KmlLocation, KmlAltitudeGeometry, KmlGeometry, KmlLod, KmlRegion, KmlLookAt, KmlScale, KmlModel, KmlCamera, KmlStyle, KmlMultiGeometry, KmlContainer, KmlObject, KmlOrientation, KmlTour, KmlDocument, KmlFeature, etc.

2.3.1 Steps for using the Google™ APIs

Step 1: Loading the Google™ Earth API:

The Browser tells Google™ to load the earth module into the google.earth namespace and specifies version.

Step 2: Creating a container for the plugin.

The Google™ Earth plugin is loaded into DIV element with a unique id.

Step 3: Creating initializing functions.

Create three functions as part of this step. In order will: (1) attempt to create a new instance of the plugin, (2) Be called when the plugin instance is successfully create, and (3) Be called if the instance cannot be created google.earth.createInstance show three options: the DIV element into which the instance should be added, the function to call when success is returned, and the function to call if a failure is returned.

The success callback function will contain all of the code required to set up all of the objects and views that will first appear when plugin instance is loaded in browser. The function must contain the GEWindow.setVisibility method, setting the window visibility to true, so that the plugin is visible inside its DIV.

Step 4: Calling the initializing function when the page is loaded.

The Google™ namespace includes the setOnLoadCallBack() function, which calls the specified function once the HTML page and requested APIs has been loaded. Using this function ensures that the plugin is not loaded until the page's DOM is completely built out.

Step 5: Loading Additional Databases.

Developer can load own database to display on the Google™ Earth globe by specifying that database during initialization, or by calling addSideDatabase() on the Earth object. Loading a database at initialization will load only that database; the standard Google™ Earth imagery will not be loaded.

Step 6: Google™ Maps Engine maps.

Google™ Maps Engine uses the term of map to refer to a collection of imagery and vector layers; it is equivalent of a database in Google™ Earth. Google™ Maps Engine maps can be loaded into the plugin using the methods above.

2.3.2 Related Features

To deploy the knowledge model on the virtual space. I applied the features for using in research below: [28-29]

Table 2.4: Mapping Knowledge Model and Google™ APIs Features.

Knowledge Model Elements	**Google™ APIs Features**
Knowledge Node	Placemark: icon, short description
Node Description	Ballon: description, link
Relationship	LineStrings
Viewpoint	Camera Control: panning, tiling - Camera - Look At Zooming Fly to
Control	Layer Controls Navigation Controls Altitude Modes
Scenario	Touring - Importing a Tour - Defining the Active Tour
Event Handling	Event Listener DOM Event
Additional	Accessors Object Containers KML *(reference in section 2.4)*

- Placemark

A point placemarks marks a position on the Earth's surface. The most basic placemark includes a standard icon and geographic location. Additionally, placemark can include: description, custom icon, and style-map that defines a rollover icon. The activities about placemarks are adding a placemark name, defining a custom icon, changing the size of an icon, and using a StyleMap to style an icon

– Balloon

Balloons are information windows displayed in Google^{TM} Earth Plugin, optionally associated with feature. Their content can include HTML, CSS, or JavaScript. Most aspects of balloons can be controlled through the API.

Feature balloons: content scrubbing such as, getDescription(), getBalloonHtml(), and getBalloonHtmlUnsafe(), HTML string balloons, HTML DIV balloons, Closing balloons, and geometries and overlays.

– Line Strings

A lineString is a connected set of line segments. The color, color mode, and width of line can all be specified in the associated placemark's LineStyle. If the line string is drawn above the Earth's surface or above the sea floor, it can be set to extend down to the terrain using the extrude property; if the line is drawn on the surface (by clamping to the ground or sea floor), it can be set to follow the terrain using tessellate property.

– Camera Control

The "view" in Google^{TM} Earth is the image user see in the plugin window; the "camera" is the viewer's location in space. These are two different ways to define a view: Camera and LookAt.

Camera: with the Camera view, the point user specify defines the location of the viewer in space. Setting a latitude and longitude moves the viewer to a specific location; the viewer can then be hoisted into space and rotated around the x, y, and z axes until the view is as desired.

LookAt: with a LookAt, the point being specified in the code is the point being viewed. Changing the range, tilt, and heading values will never change the absolute point in space at which the view is aimed.

Getting the current view: Developer can use the copyAsLookAt() function to return the latitude and longitude of the point at which the camera is looking, the altitude at which the camera is set, its tilt, and the compass heading in degrees.

Panning the camera - The camera can be moved to either an absolute location, defined by a Camera or LookAt, or to a position relative to its current view.

Panning to an absolute location - Locations on the globe are specified by their latitude and longitude values. The Google^{TM} Earth Plugin accepts values with up to six digits of precision (0.123456). To move from the current view to an absolute location, while retaining the tilt, range, heading, and altitude values: Panning relative to the current view - The view can be moved in by direction relative to its current position.

Tilting the camera - A LookAt can contain a tilt value between 0 and 90 degrees inclusive, with 0 being directly above the viewed point, and 90 viewing along the horizon. A Camera can contain a tilt value between 0 and 360 degrees. 0 degrees is a view directly down from the specified point; 90 set a view along the horizon; 180 views directly up into the sky. In addition, a Camera accepts a roll attribute, which rotates the viewer around the z axis.

– <u>Zooming</u>

Zooming in and out is controlled by the range attribute for a LookAt, and the altitude attribute for a Camera. Changing the altitude attribute of a LookAt changes the altitude of the point being viewed. Because the viewer range is relative to this point, the viewer's altitude is also changed.

– <u>Fly To</u>

Developer can control the speed at which the plugin moves to newly-specified locations. GEOptions.setFlyToSpeed() accepts a float from 0.0 to 5.0, inclusive, as well as SPEED_TELEPORT. Speed increases with the value; SPEED_TELEPORT moves to the specified location instantly.

– <u>Layer and Controls</u>

Layers contain additional information that is draped over the base Google™ Earth imagery layer. Navigation controls allow the user to pan, tilt, and zoom using controls superimposed on the viewport. By default, the terrain layer is the only one layer displayed when the Google Earth Plugin first loads. Not all of the layers that are available in the Google™ Earth desktop client are available with the plugin.

– <u>Navigation Controls</u>

Navigation controls allow a user to move around in Google™ Earth, and include zoom, pan, tilt, scroll, and rotate controls. These can be always visible, never visible, or can be hidden until the user moves their mouse cursor over the controls' area in the plugin.

The navigation controls can be positioned at any corner of the Google™ Earth Plugin window. To do so, change the value of the x and y units to be relative to either the top or bottom of the window, and either the left or right.

– <u>Altitude Modes</u>

Absolute altitude modes are calculated from sea level. To place an object underwater, specify a negative altitude value. A negative altitude over land will place the object below the Earth's surface. Objects below the Earth's surface cannot be viewed or clicked.

– <u>Touring</u>

The Google™ Earth Plugin can play tours authored in KML, allowing viewers to interact with the Earth environment while viewing scripted controlled tours. The plugin currently exposes tour playback methods; tour authoring is not supported in the plugin.

Importing a tour

Simple tour: with the <gx:Tour> feature as the root-level feature of the KML, can be fetched and passed directly to GETourPlayer. The tour must be the only feature in the KML file.

Complex tour: If developer tour is contained within a KMZ file, or if developer KML file contains more than just a tour, or if the tour is nested within a container (such as <Document> or <Folder>), developer will need to manually look for the <gx:Tour> feature within the file. One way is to "walk" through the file's DOM until the KML Tour feature is found. Developer can use the kmldomwalk.js utility script to do this. For an example, the KML file includes some placemarks and features, which need to be loaded into Earth as well. The preceding code simply loads the fetched tour into the Earth plugin. It does not include controls for setting the fetched tour as active, or for controlling playback.

Defining the Active Tour: Any number of tour objects can be loaded into the plugin, but only one can be set as the currently active tour. Once a tour is set as active, the tour controls appear on the screen.

– Events

The Google™ Earth API provides a number of different events, which can be used with google.earth.addEventListener to provide additional interactivity in developer applications. Using event listeners, developer can create actions that are triggered on mouse events (such as clicks, movement, or dragging) or screen events (such as change to view).

Adding an Earth API event listener requires three arguments, and accepts an optional fourth: the object on which to add the listener, the even to listener for, the function to call when the event is fired, and (optionally) whether or not this listener should initiate capture (refer to relevant W3C DOM documentation for details of event capture). The default value for this fourth argument is false.

– Event Listeners

Mouse events can be attached to most geometry in the plugin (the exception is 3D models) to entire viewport, or to the globe only. There are listeners for most mouse events, including clicks and movements. For a full list of mouse events, refer to the GEEventEmitter_Interface_Reference.

View events are fired when the view begins to change, while it is changing, and when it has ended. Listeners for view events must be attached to the viewport object of the plugin instance (ge.getView()).viewchangeend may fire in the middle of a view change, if the plugin pauses for a brief period during the change. If users are relying on viewchangeend to indicate the absolute end of a view change, it is recommended that user include a brief timeout to ensure that no further view changes are to follow

A *frameend event* is fired when Earth has finished rendering the viewport. This event will be called many times in succession when the viewport is changing. Add a listener for this event and make incremental changes to the viewport for smooth animation. A frameend listener must be attached to the Google™ Earth Plugin instance.

The *balloonclose event* is fired when the current description balloon is closed. Its listener must be attached to the plugin instance.

Removing event listeners

Developer can remove event listener using removeEventListener(). Developer must pass the same object, event type, and function name to removeEventListener() as were specified when creating the event listener.

– DOM Events

To add listeners to HTML elements on the page outside of the plugin, developer can use this helper function that will work across all modern browsers: This accounts for the different methods of attaching handlers in Internet Explorer and most other browsers; Internet Explorer® uses attachEvent and onclick, while other use addEventListener and click.

– Accessors

getUrl() : An object's URL is its base address concatenated with its ID using the # character. This URL can be returned by calling getUrl() on the object. Objects created with the API (rather than imported as KML) do not have a base address; their URL will consist of a # character and their ID.

getComputedStyle() : Returns the object's style properties as a KML style object, merging any inline styles with styles imported from setHref() or a StyleUrl.

getElementsByUrl() : Objects that are imported as KML have an identifying URL consisting of their base address and ID, joined with the # character.

getElementById() : When an object is created with the API, rather than imported as KML, the object does not have a base address. In this case, the object can be returned by passing only its ID to getElementById().

getElementByType() : Developer can obtain an array of all elements of a certain type, by passing that type as a string to getElementsByType().

– Object Containers

There are a number of container objects in the Google™ Earth API. These are used to hold arrays of related objects:

- A GELinearRingContainer holds an array of linear ring objects. For example, a polygon's inner boundaries are stored in a linear ring container.
- GEFeatureContainers contain features, as with folders in KML.
- GEGeometryContainers hold any number of geometries in a MultiGeometry object.
- Containers are abstract classes and cannot be created directly from the API.

Modifying Container: Containers have methods that allow developer to enumerate, add, remove, and manipulate individual items in the collection. Some common methods are described below; for a full list, refer to the API Reference for the applicable container type.

2.4 Keyhole Markup Language (KML)

Keyhole Markup Language (KML) is an open-standard markup language for display of geographic data in geo-browser. The Google™ Earth Plugin can import KML in different ways, in order to display features, tours, and views. The plugin supports all KML 2.2 tags, as well as extensions to KML using the gx namespace. The plugin can also return KML representations of features, whether those features were imported as KML or created with the APIs.[30]

2.4.1 Importing KML

There are three methods of importing KML into the plugin.

(1) KmlNetworkLink loads a KML or KMZ file from a specified URL. The fetched KML can then be added to the plugin just as any other object, using ge.getFeatures().appendChild(). KmlNetworkLink references a KML or KMZ file by its URL. The contents of the file can be added into the Earth instance, but they cannot be manipulated before being displayed – there is no access to the KML's DOM. A NetworkLink is a standard way of importing content in the KML language.

fetchKml and parseKml accept KML in different ways, but both return a KmlFeature object, allowing for access to, and manipulation of, the object's hierarchy (its children and other descendant features).

(2) fetchKml also loads KML from a URL, but returns a KmlFeature object representing the root KML feature. The object's KML DOM can be accessed and updated before it is appended to the plugin's KML DOM. fetchKml calls specified callback function when a success or failure message is returned. fetchKml's asynchronous behavior allows the plugin to continue while the network fetch is completing.

(3) parseKml takes a KML string, and also returns a KmlFeature object. As with fetchKml, the returned object's KML DOM can be accessed. parseKml returns the object immediately.

<u>Accessing and Modifying the DOM</u>

The fetch KML object's DOM can be accessed and modified before or after the KML features are displayed. In the example below, a placemark is created using the Earth API, and is then appended to the KML object returned by the fetchKml() or parseKml() function. When the KML object is added to Earth, the API-created placemark is included.

<u>Getting a feature's KML representation</u>

The Google™ Earth Plugin can provide a feature's KML representation, so that the feature can be displayed in any KML-compatible application. To obtain a feature's KML, use KmlFeature.getKml().

2.4.2 Server-side and Client-side KML Rendering

When KML content containing a large number of features (placemarks, polygons, lines) is loaded into Google™ Maps, a server-side decision is made about whether or not to render the document's features on the server or in the user's web browser. The user experience, including feature appearance and interactivity, will not be affected by this decision; the content will look and behave the same way, regardless of the chosen method. However, there is a subtle difference in the behavior of these methods that surfaces when using the GGeoXml class to render KML content in the Google™ Maps API. When simple KML files are loaded via GGeoXml and client-side, browser-based feature rendering is chosen; Goverlay-based objects are created and added to the map, triggering addoverlay events accordingly. On the other hand, when large KML files are loaded with GGeoXml and server-side rendering is invoked, these objects are not created and addoverlay is not triggered. For this reason, it is not recommended to rely on the addoverlay event being triggered on individual KML features loaded via GGeoXML.[31]

2.4.3 Creating and Sharing KML Files

Developer can create KML files with the Google^{TM} Earth user interface, or can use an XML or simple text editor to enter "raw" KML from scratch. KML files and their related images (if any) can be compressed using the ZIP format into KMZ archives. To share KML and KMZ files, developer can e-mail them, host them locally for sharing within a private internet, or host them publicly on a web server. Just as web browsers display HTML files, Earth browsers such as Google^{TM} Earth display KML files. Once developer have properly configured server and shared the URL (address) of KML files, anyone who is installed Google^{TM} Earth can view the KML files hosted on developer public web server. Many applications display KML, including Google^{TM} Earth, Google^{TM} Maps, Google^{TM} Maps for mobile, NASA WorldWind, ESRI ArcGIS Explorer, Adobe Photoshop, AutoCAD, and Yahoo! Pipes.

KML is an open standard official named the OpenGIS® KML Encoding Standard (OGC KML). It is maintained by the Open Geospatial Consortium, Inc. (OGC). Figure 2.6 shows the diagram of object oriented hierarchy related with KML elements. They are a useful way for a single element to serve as the programmatic foundation for multiple similar derived elements. All elements derived from *Object* can have id assigned to them. This id is used by the KML update mechanism for files loaded with a NetworkLink. It is also used by shared styles. The id is a standard XML ID. Because KML is an XML grammar and file format, tag names are case-sensitive and must appear exactly. When developers are editing KML text files, developer can load the schema into any XML editor and validate KML code with it. [32]

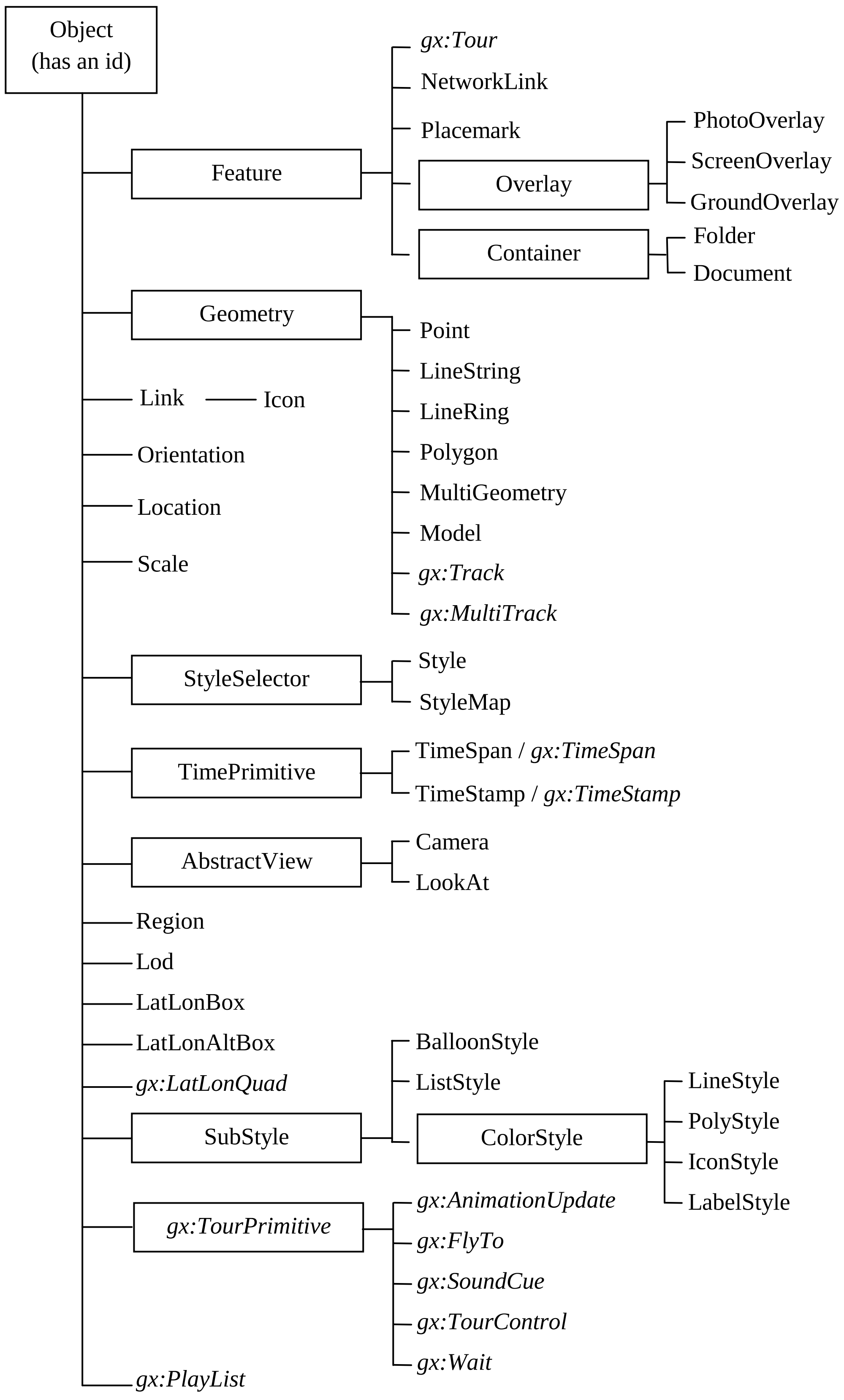


Figure 2.6: Diagram of Object Oriented Hierarchy Related of KML Elements.

Bibliography

[1] Mark A. Musen, L. M. Fagen, D. M. Combs, and E. H. Shortliffe, "*Use of a Domain-Model to Drive an Interactive Knowledge-Editing Tool*", Int. J. Man-Machine Studies, vol.26, pp.105-121 (1987).

[2] Chaomei Chen and Ray J. Paul, "*Visualizing a Knowledge Domain's Intellectual Structure*", IEEE (2001).

[3] Chau, K.W., Chuntian, C. and Li, C.W., "*Knowledge Management Systems on Flow and Water Quality Modeling*", Expert System with Application, vol.22, pp.321-330 (2002).

[4] Chen-Burger, J., "*Knowledge Sharing and Inconsistency Checking on Multiple Enterprise Models*", Informatics Research Report EDI-INF-RR-0037, AIAI, University of Edinburgh (2001).

[5] Gao, F, Li M and Nakamori, Y., "*System Thinking on Knowledge and its Management: Systems Methodology for Knowledge Management*", Journal of Knowledge Management, vol.6, no.1, pp.7-17 (2002).

[6] Hendriks, P. and Virens, D., "*Knowledge-based Systems and Knowledge Management: Friends or Foes?*", Information & Management, vol.35, pp.113-125 (1999).

[7] Loucopoulos, P. and Kavakli, V., "*Enterprise Knowledge Management and Conceptual Modeling*", Lectures Notes in Computer Science, vol.1565, pp.123-143 (1999).

[8] Sallis,E. and Jones, G., "*Knowledge Management in Education: Enhancing Learning & Education*", Kogan Page, London (2002).

[9] Studer, R., Benjamins, V.R.,and Fensel, D., "*Knowledge Engineering: Principle and Methods*", Data & Knowledge Engineering, vol.25, pp.161-197 (1998).

[10] Visser, P., Kralinger, R. and Capon, T., "*A Method for the Development of Legal Knowledge Systems*", ICAIL'97, Melbourne, Australia, pp.151-160 (1997).

[11] Wigg, K.M., "*Knowledge Management: Where Did It Come From and Where Will It Go?*", Expert System with Application, vol.13, no.1, pp.1-14.

[12] Choo, C.W., "*Working with Knowledge: How Information Professionals Help Organization Manage What They Know*", Library Management, vol.21, no.8, pp.395-403 (2000).

[13] Grosso, W.E., Eriksson, H. Fergerson, R.W., Gennari,J.H., Tu, S.W. and Musen, M.A., "*Knowledge Modeling at the Millennium (The Design and Evolution of Protégé-2000)"*, SMI Technical Report-SMI-1999-0801.

[14] Motta, E., "*Reusable Components for Knowledge Modeling: Case Studies in Parametric Design Problem Solving*", IOI Press, Amsterdam.

[15] Rolland, C. and N. Prakash, "*A Proposal for Context-Specific Method Engineering*", Atlanta, GA, USA, pp.191-208 (1996).

[16] Henderson-Sellers, B., "*Method Engineering for OO System Development*", Comm. ACM, vol.46, no.10, pp.73-78 (2003).

[17] Tiwana, A., and Ramesh, B., "*A Design Knowledge Management System to Support Collaborative Information Product Evolution*", Decision Support Systems, vol.31, pp.241-262 (2001).

[18] Weilinga, B., Sandberg, J. and Schreiber, G., "*Methods and Techniques for Knowledge Management: What Has Knowledge Engineering to Offer?*", Expert System with Application, vol.13, no.1, pp.73-84.

[19] Bloodgood,J.M. and Salisbury, W.D., "*Understanding the Influence of Organizational Changes Strategies on Information Technology and Knowledge Management Strategies*", Decision Support Systems, vol.31, pp.55-69 (2001).

[20] Busse S., Kutsche R., Lesser U., Weber H., "*Federated Information Systems: Concepts, Terminology and Architectures*", Technical report 99-9. TU Berlin, April.

[21] Brinkkemper, S., "*Method Engineering: Engineering of Information Systems Development Methods and Tools*", Inf. Software Technology, 38(4), pp.257-280 (1996).

[22] Nuseibeh, B., "*Towards a Framework for Managing Inconsistency between Multiple Views*", SIGSOFT 96 Workshop, San Francisco, pp.184-186 (1996).

[23] Jacobson I., Christerson M., Johnsson P., Overggard G., "*Object-Oriented Software Engineering, A Use Case Driven Approach*", Addison-Wesley/ACM Press, Wokingham, England (1992).

[24] Awad, E.M., "*Building Expert Systems: Principles, Procedures, and Applications*", Minneapolis: West Publishing Company (1996).

[25] G. Booch, J. Rumbaugh, and I. Jacobson, "*The Unified Modeling Language User Guide*", Addison-Wesley, Reading, MA (1998).

[26] Scott, K., "*UML Explained*", Addison Wesley, Reading, Massachusetts (2001).

[27] Google^{TM} Earth API, https://developer.google.com (2012).

[28] Hao Zhang, Manchun Li *et al.* : "*Land use information release system based on Google Maps API and XML*", Geoinformatics 18th Intl. Conf., Beijing, China (2010).

[29] Jiang Peng, Xu Huaiyu *et al.* : "*An instant and interactive platform based on Google Earth Plug-in*", Advanced Computer Control (ICACC) 2nd Intl. Conf. , Shenyang, China (2010).

[30] Open Geospatial Consortium Inc. : "*OpenGIS® Catalogue Services Specification 2.0.2 – ISO Metadata Application Profile*", Version 1.0 (2007).

[31] The Open Geospatial Consortium Inc.: "*OpenGIS KML Encoding Standard (OGC KML)*".

[32] M. R. Genesereth and R. E. Fikes, "*Knowledge Interchange Format Version 3.0 Reference Manual*", Report Logic 92-1, Logic Group, Stanford University, California (1992).

Chapter 3

Knowledge Schemas

This chapter describes about knowledge schemas that designed for this system. Section 3.1 describes concept of knowledge model in CommonKADS and category of knowledge structure: Domain knowledge in 3.1.1, Inference knowledge in 3.1.2, Task knowledge in 3.1.3, and comparison with others analysis approaches in 3.1.4. For section 3.2 describes about an organization model and related models. Section 3.3 describes a schema name Knowledge Landscape which designed for concept model. Section 3.4 describes about Knowledge Atlas schema which designed for context model. Section 3.5 describes the Knowledge Schema. Finally, section 3.6 is conclusion.

3.1 Knowledge Model

Knowledge thus typically has an "aboutness" character: it tells us about the way to understand some other piece(s) of information. The knowledge model specifies the knowledge and reasoning requirements of the prospective system. It is a tool that helps us clarifying the structure of a knowledge-intensive information-processing task.[1] The different between a knowledge system and database application: Database application seldom interested in the actual facts that have to be placed in the database. For the knowledge system, a knowledge base typically contains certain pieces of knowledge such as rules, which are of interest. In knowledge modeling, we typically distinguish multiple knowledge bases containing different types of knowledge (e.g., instances of different rule types). [2]

CommonKADS moves away from the idea of one large knowledge base. Instead, we need to identify parts of the knowledge base in which the knowledge fragments (e.g., rules) share a similar structure, as shown in figure 3.1.

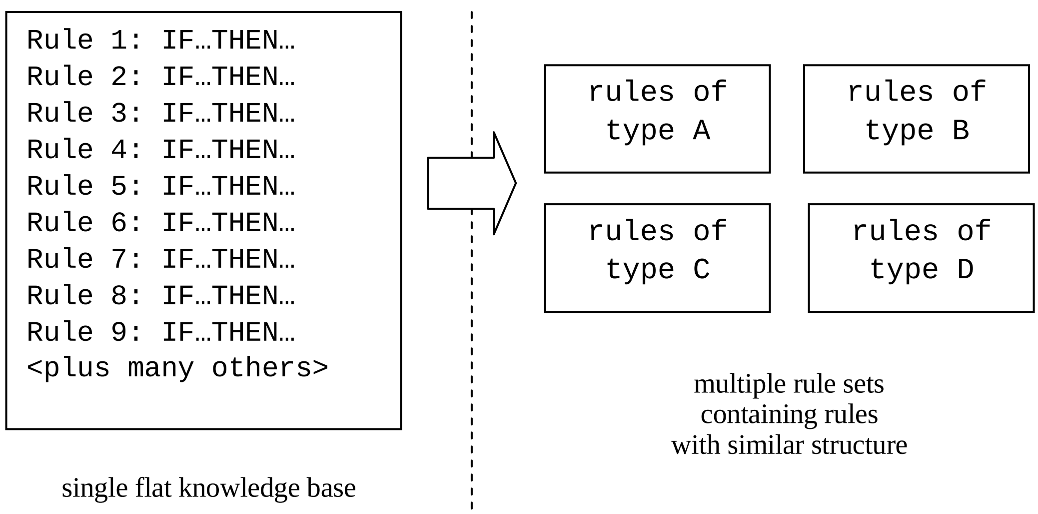


Figure 3.1: Knowledge Fragments (e.g, rules) Share a Similar Structure.

Knowledge modeling is a technique that helps to clarify the structure of a knowledge-intensive task and provides a specification of the data and knowledge structures that required for the application. It is developed as part of the analysis process therefore phrased in the vocabulary of the application meaning both of domain and reasoning task. In CommonKADS methodology, knowledge model consists of three parts each capturing a related group of knowledge structure called “knowledge category”.[3] It consists of: domain knowledge, inference knowledge and task knowledge.

3.1.1 Domain Knowledge

This category specifies the domain specific knowledge and information types. Its modeling implies capturing the static structure of information and knowledge types. Just like in regular data modeling, a schema is constructed containing the major types and relations occurring in an application domain. The notation used is similar to a UML’s class diagram principle but no included behavioral things such as, operations and methods. Domain knowledge description typically consists of two types: Domain schema and Knowledge base. [4]

Domain schema is a schematic description of the domain specific knowledge and information through a number of type definitions. From a general software engineering point of view, the domain schema resembles a data model or object model. In addition to UML class diagram, constructs are included to cover modeling aspects that are specific to knowledge-intensive systems. In practice, the three main modeling construct are CONCEPT, RELATION, and RULE-TYPE. In addition, several other constructs are available such as SUPER/SUBTYPE OF and AGGREGATE/PART. All of concepts similar to UML class model. Figure 3.2 shows the domain schema in CommonKADS methodology.

```
CONCEPT name;
     ATTRIBUTES:
     value: dial-value;
END CONCEPT name;

VALUE-TYPE dial-value;
     VALUE-LIST: {zero, low, normal};
     TYPE: ORDINAL;
END VALUE-TYPE dial-value;
```

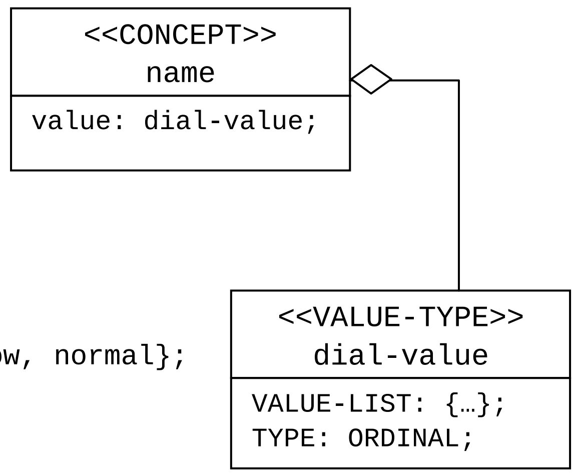


Figure 3.2: Sample of the Domain Schema in CommonKADS Methodology.

Knowledge base contains instances of the types specified in domain schema. It contains certain pieces of knowledge such as rules which are of interest. In knowledge modeling, we distinguish multiple knowledge bases containing different types of knowledge such as, instances of different rule types. Figure 3.3 shows the sample of knowledge base in CommonKADS methodology.

```
KNOWLEDGE-BASE knowledge-base_name;

      USES:

            <RULE-TYPE> FROM <SCHEMA NAME>;

            …

      EXPRESSIONS:

      /*RULE-TYPE EXPRESSION*/

      antecedent   expression   CONNECTION   SYMBOL   consequent
      expression

      …

END KNOWLEDGE-BASE knowledge-base_name;
```

Figure 3.3: Sample of the Knowledge base in CommonKADS Methodology.

Domain Knowledge Lifecycle The separation of “domain schema” and “knowledge base” means that we have to reinterpret the term “knowledge acquisition” as consisting of at least two steps: (1) Defining a knowledge type such as a rule type, (2) Elicit the instances of this type and putting them in a knowledge base.

3.1.2 Inference Knowledge

The inference knowledge describes the basic inference steps that want to make using the domain knowledge. It describes how these static structures can be used to carry out a reasoning process. In software engineering terms, the inferences represent the lowest level of functional decomposition. The components of the inference knowledge are inference, knowledge role, and transfer function. [5]

Inferences are best seen as the building blocks of the reasoning machine because it carries out a primitive reasoning step. Typically, an inference uses knowledge contained in some knowledge base to derive new information from its dynamic input. In addition, inferences are indirectly related to the domain knowledge. The indirect coupling of inference and domain knowledge enables to reuse inference descriptions independently from domain knowledge.

Specification of inference: The main feature that distinguishes an inference from a traditional “process” or “function” is the way in which the data on which the inference operates are described. Inference I/O is described in term of functional roles: abstract names of data objects that indicate their role in the reasoning process. Figure 3.4 shows the inference knowledge and domain knowledge mapping with the knowledge role.

```
KNOWLEDGE-ROLE  dynamic_input_role;
      TYPE: DYNAMIC;
      DOMAIN-MAPPING:  concept_name;
END KNOWLEDGE-ROLE dynamic_input_role;
```

```
KNOWLEDGE-ROLE dynamic_output_role;
      TYPE: DYNAMIC;
      DOMAIN-MAPPING:  concept_name;
END KNOWLEDGE-ROLE dynamic_output_role;
```

concept_name from inference-domain mapping

Figure 3.4: Mapping of Inference and Domain with Knowledge Role.

Knowledge role (Inference I/O) is described in terms of functional role: abstract names of data objects that indicate their role in the reasoning process. It has two types of roles: dynamic and static role.

- *Dynamic roles* are the run-time inputs and outputs of inferences. Each invocation of the inference typically has different instantiations of the dynamic roles.

- *Static roles* are more or less stable over time. Static roles specify the collection of domain knowledge that is used to make the inference.

Transfer function is a function that transfers an information items between the reasoning agent that described in knowledge modeling and its environment such as another system or some users. Transfer functions are black boxes from the knowledge model point of view: only their name and I/O are described. Transfer function has 4 functions: obtain, receive, present, and provide.

- *Obtain*: The reasoning agent requests a piece of information from an external agent. The reasoning agent has the initiative. The external agent holds the information item.

- *Receive*: The reasoning agent gets a piece of information from an external agent. The external agent has the initiative and also holds the information item.

- *Present*: The reasoning agent presents a piece of information to an external agent. The reasoning agent has the initiative and also holds the information item.

- *Provide*: The system provides an external agent with a piece of information. The external agent has the initiative. The reasoning agent holds the information item.

	System Initiative	*External Initiative*
External Information	Obtain	Receive
Internal Information	Present	Provide

Figure 3.5: Type of Transfer Function in Inference Knowledge.

In CommonKADS methodology, the set of inference steps can be represented graphically in CommomKADS's inference structure. The combined set of inference specifies the basic inference capability of the target system. It is an abstract representation of the possible steps in the reasoning process. For model the inference, there use a specific notation that no direct UML equivalent to develop their structure. [6] Figure 3.6 shows a CommonKADS's inference structure and its notations.

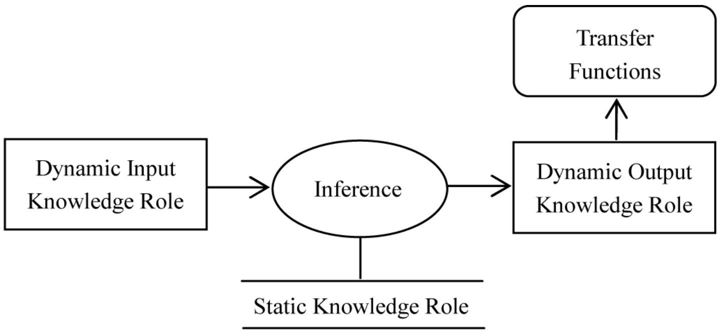


Figure 3.6: Example of CommonKADS's Inference Structure.

3.1.3 Task Knowledge

The third category of knowledge modeling is task knowledge. Task knowledge describes what goals and application pursues, and how these goals can be realized through decomposition into subtasks and ultimately inferences. Task knowledge is described in a hierarchical fashion: top-level tasks are decomposed into smaller tasks, which in turn can be split up into even smaller tasks. Task knowledge is similar to the higher levels of functional decomposition in software engineering, but also includes control over the functions involved. Task can be decomposed into subtasks or into basic inferences. At the lowest level of task decomposition, the tasks are linked to inferences and transfer functions.[7]

The TASK and TASK-METHOD can best be understood as respectively the "what" view (what needs to be done) and the "how" view (how is it done) on reasoning tasks. In most real-life models, one level of decomposition is insufficient. In that case, a top-level task is decomposed in several new tasks, which again are decomposed through other methods, and so on. Tasks that are not decomposed further into other tasks are called primitive tasks; the other tasks are called composite tasks. Tasks are divided into subtasks up to level of elementary inferences that are not decomposed further. As a result, a task is composed of a number of combined inferences yielding an inference diagram.[8]

3.1.4 Comparison with Other Analysis Approaches

Difference 1: "data model" contains both data and knowledge. Knowledge can be seen as "information about information". It implies that parts of the "data model" describe how we should interpret or use other parts. We could also want to describe a domain-knowledge type that allows us to infer the latter from the former. This requires specialized modeling tools, in particular the construct RULE-TYPE.[9]

Difference 2: "functions" are described datamodel-independent. Decoupling of functions and data makes a knowledge model more complex, but it enables exploitation on powerful forms of reuse. The input/output of functions in a knowledge model is not described in terms of data model elements, but in terms of task-oriented "role" names. These "roles" act as placeholders for data-model elements. Effectively, role decouples the description of the static information structure on the other hand and the functions on the other hand.

Difference 3: the need to represent "internal" control. In OMT, control is specified through state-transition diagram, useful for systems in which information processing is mainly driven by external events. However, in reasoning tasks, there is usually a clear need to also represent the internal control of the reasoning.

Difference 4: knowledge model abstracts from communication aspects. The knowledge model abstracts from all issues concerning interaction with the outside world. These interactions are described in the communication model.[10]

Figure 3.7 shows the schematic view of the data-function debate comparing between object-oriented analysis and structured analysis, including their view point with the CommonKADS methodology.[11]

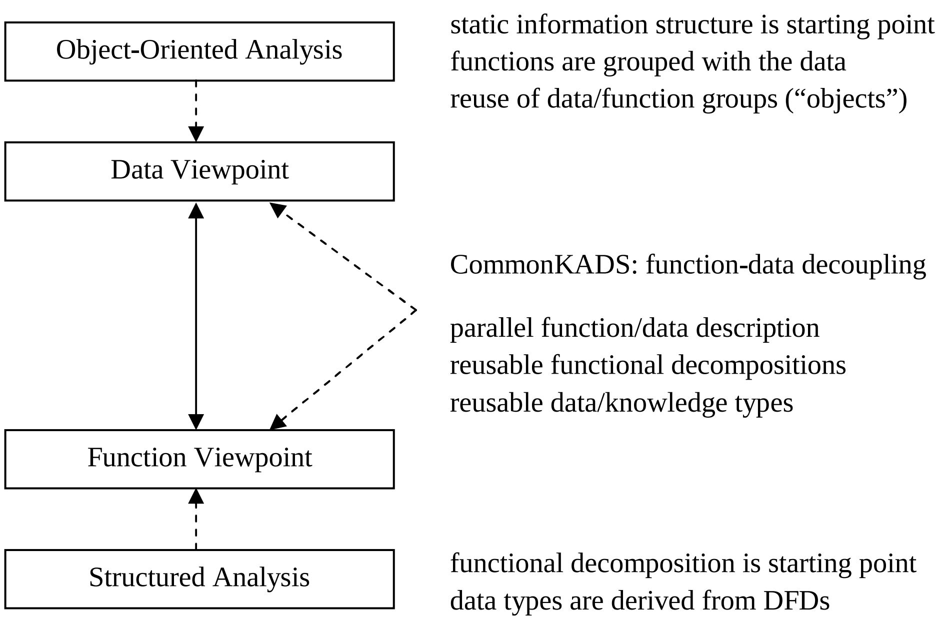


Figure 3.7: Schematic View of the Data-Function Debate.

In the Yourdon approach, functional decomposition is the starting point of analysis; in the modern object-oriented approaches the “data” are the initial focus of attention. CommonKADS takes an intermediated position, assuming both data and function descriptions can be stable and reusable.[12]

3.2 Organization Model

The CommonKADS approach intentionally combines and integrates ideas coming from various areas in organizational analysis and business administration. It has been influenced by soft systems methodology, especially in its thinking on how to come a clear and agreed picture of what the real problems and opportunities in an organization are. In this regard, it is also useful to consult literature on organizational learning.

One of the prominent tools in managing knowledge is the knowledge-based systems. It can be deployed as the technological means for capturing and managing both of tacit and explicit knowledge as part of an organization-knowledge management initiative.[13] The capabilities of knowledge-based systems were no longer limited to the emulation of expert reasoning; they could also be applied to managing organization knowledge such as business rules, procedures and guidelines.

From the study on assessing the knowledge-based systems of Gill’s problems[14], the successful adoption of knowledge systems is not primarily dependent on either technical or economic reasons. It depends on mainly due to organizational and managerial issues. From this study, I raise two issues concerns in an organizational aspect. The first concern is a coordination of knowledge system development with organization’s business and IT strategies. Knowledge system should be able to support the strategic information system needs and overall business processes. The second concern about failure to understand the task that system would best support. Generally, not all tasks can be performed better by the system. There are some tasks better performed by human especially when the domain task is multidimensional and requires complex judgments.

From above concerns, the CommonKADS methodology provides tool for scoping and feasibility analysis for the organizational aspects. CommonKADS aims to integrate organization process analysis and information analysis. It provides worksheets to describe the organizational context, the performed-tasks, and the responsive agents.[15] For example, Table 3.1 shows a sample of worksheet suite. This worksheet is OM-4: Knowledge assets worksheet.

Table 3.1: Sample of a Worksheet in Organization Model.

Organization Model		**Knowledge Assets Worksheet OM-4**				
KNOWLEDGE ASSET	POSSESSED BY	USED IN	RIGHT FORM?	RIGHT PLACE?	RIGHT TIME?	RIGHT QUALITY?
Concept development and testing: ice-cream processing	Manufacturing	2. Feasibility phase	Yes	No (needed at Development)	Yes	No (incomplete, heuristic)
...	...	...	...	...	...	...
Concept development and testing: finished product specification	Development core team	2. Feasibility phase	No: paper form too limited	Yes	Yes	Yes

To develop knowledge-based systems, knowledge engineers collect the system information from organization aspect by the suite of worksheets. All of worksheets are separated in three groups and one checklist: Organization model (OM), Agent model (AM), Task model (TM), and Organization-Task-Agent Checklist (OTA).[16]

Organization model is regarded as a feasibility study for knowledge system. The study is conducted based on problems and opportunities of system. It focuses on such areas as, structure, process, people, resources, process breakdowns and knowledge assets. This model has three main proposes: (1) To identify an area in organization where knowledge-based applications can be implemented, (2) To identify what impact the knowledge-based application will have in organization when it is implemented, and the last, (3) To provide the system developers with a feeling for where the organization the applications will be deployed.[17]

For Agent model, it has purpose to understand a role played by different agents when performing a task. In knowledge system, agents can be as people, computers or any other entity that can perform the task. In model worksheets, they specify agent characteristics, authority to perform the task and any associated constraints.

Last model is the Task model. It has purpose to provide an insight of impact that introducing the knowledge system will have on organization. This model refers to characteristics of the business processes, for examples, inputs and outputs, pre-conditions, performance and quality, function of the agents that will carry out the processing, flow of knowledge between agents and their overall control, the knowledge and competences of the agents and the resources available to deliver the business process.[18]

In this research focused on the organization model, I applied the organization model from the worksheet to the model using the UML extension mechanism. I gathered all of organization model worksheet and conclude to the one metamodel.

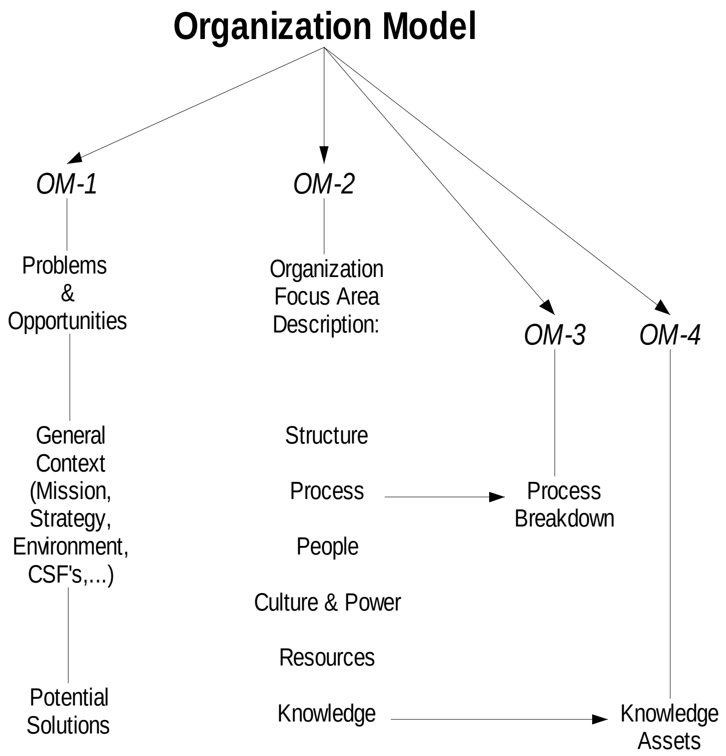


Figure 3.8: Roadmap of Models in CommonKADS Context Level.

Figure 3.8 shows a road map for carrying out knowledge-oriented organization and task analysis in context level of the CommonKADS methodology. In the organization model, it consists of four worksheet that investigate rely on their goal. All of worksheets have relationship and can combine to one model with their relationship.[19]

Table 3.2: Context Level Models and Description.

Model	Description
	Organization Model
OM-1	Define problems and opportunities.
OM-2	Describe organization aspects.
OM-3	Describe all business process.
OM-4	Describe Knowledge assets of OM-2.
OM-5	Represents a big picture of all benefits versus the cost and needed technologies for the solution.
	Task Model
TM-1	Refinement of data in OM-3 (Business process).
TM-2	Refine model of OM-4 (Knowledge assets).
	Agent Model
AM-1	Agent description.

In table 3.2 shows the worksheets number and their description. Figure 3.9 shows the set of worksheets structure in CommonKADS methodology (organization model) and their description.[20]

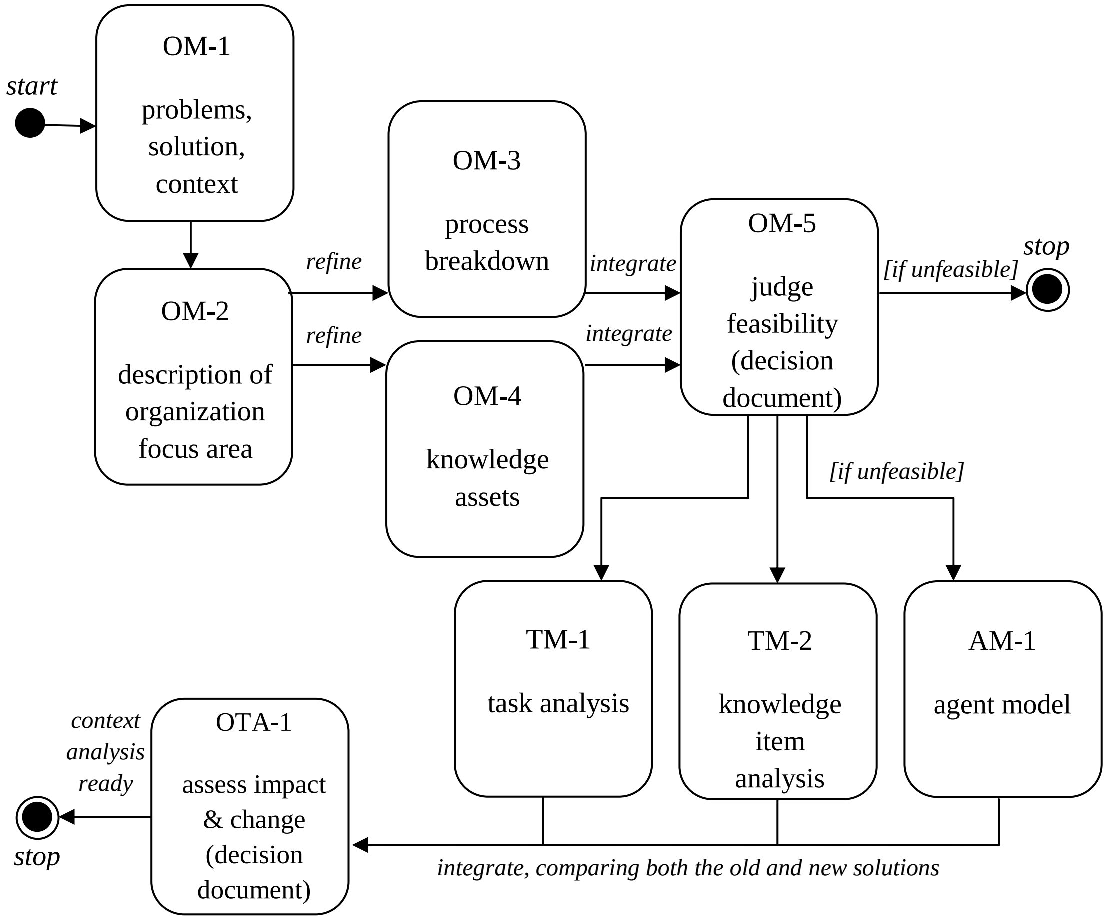


Figure 3.9: Set of Worksheets Structure in Context Level.

3.3 Knowledge Landscape Schema

To model the knowledge modeling, it composes of two levels point of views: architectural level and metaclass (component) level. Architectural level is a higher level that describes the structure of knowledge model in terms of package dependencies, as well as the control regimen through which these packages interact. Architectural level composes with three main packages based on the knowledge category in CommonKADS. It consists of domain knowledge package, inference knowledge package and task knowledge package. Inside of domain knowledge package, it has two sub-packages: knowledge schema package and knowledge base package. Figure 3.10 shows an architectural view of knowledge model and the peripheral packages that related with knowledge modeling.

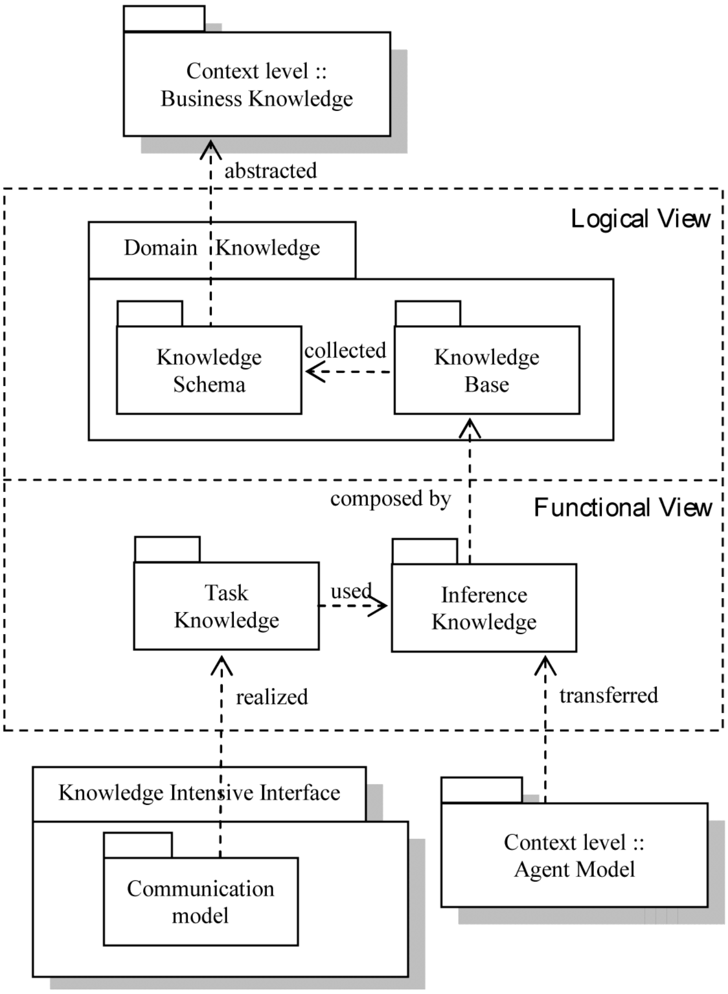


Figure 3.10: Architectural View of Knowledge Model in CommonKADS Concept.

From figure 3.10, the knowledge model architecture has two perspective views: logical view and functional view. The logical view provides abstract for represent the domain knowledge from the knowledge domain-sources, such as business knowledge in an organizational aspect. On the other hand, the functional view realizes the scenarios from the knowledge-intensive task that correspond with inference and task knowledge in object’s form, by inside-out and outside-in realized techniques. Descriptions of architectural elements are itemized in Table 3.3.

Table 3.3: Architectural Knowledge Modeling Concept.

Package Stereotype	**Responsibility**
	Logical view
Domain Knowledge	Model the domain specific knowledge and information types.
Knowledge Schema	Describe domain specific knowledge through a number of type definitions.
Knowledge Base	Contain instances of the types specified in domain schema.
	Functional view
Inference Knowledge	Specification of invocation of an inference method.
Task Knowledge	Model of the reasoning function.

Figure 3.11 shows the metaclass diagram in component level. It defines the knowledge model with the UML extension mechanisms and describe knowledge model characteristic in an object oriented approach.

Table 3.4: Component Knowledge Modeling Concept.

Component Stereotype	**Responsibility**
Concept	Class that represents the category of things.
Relation	Used for more complicated types of modeling and defined through a specification of arguments e.g. inheritance and aggregation relationship etc.
Rule type	Categorization and specification of domain knowledge.
Knowledge Base	Collection of data stores that contains instances of domain knowledge types.
Inference	The lowest level of functional decomposition on carrying out primitive reasoning steps.
Role	Defines functional roles in reasoning process.
Dynamic knowledge role	Run-time inputs and output of an inference.
Static knowledge role	The collection of domain knowledge is used to make the inference.
Transfer function	Transfers information between Inference knowledge and the reasoning agent / external entities.
Task	Defines the reasoning function and invokes the corresponding task method.
Task Method	Formalize method control structure in control language provided by the architecture.

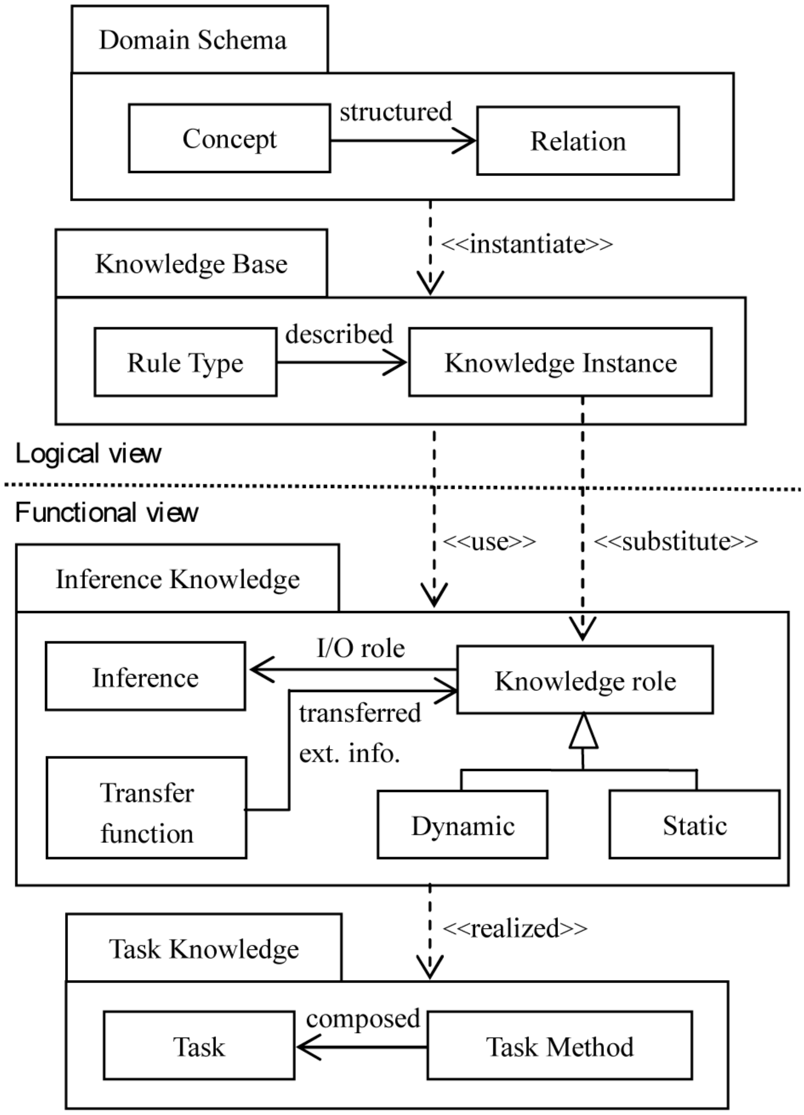


Figure 3.11: Metaclass Diagram of Knowledge Model with UML Extension.

Concepts of component elements in metaclass diagram are itemized in Table 3.4. Figure 3.12 is an example schema from the case-study “online course registration portal”. The domain concerns the online course registration process in which courses are registered for the students in the respective LP/Semester. On case of knowledge intensive task, I have mainly focused concern with the “student counseling” for selection of courses of interest.

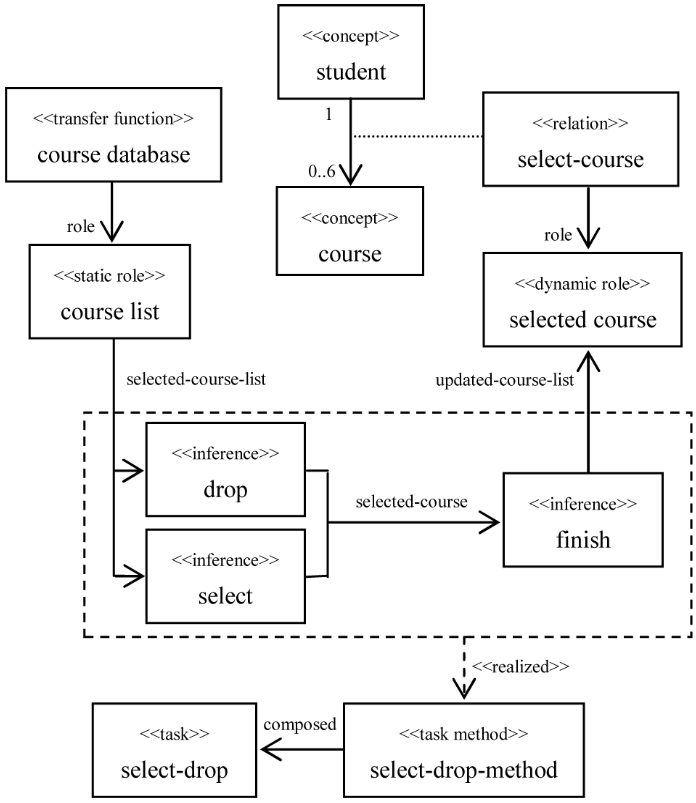


Figure 3.12: Knowledge Landscape in Online Course Registration Portal case study.

3.4 Knowledge Atlas Schema

Although different organization systems have different goals and internal structures, they use similar concepts to describe their structure and operations. To model knowledge atlas with UML extension mechanism in high level, it consists of three package extensions rely on concept of context model in CommonKADS methodology: organizational aspect package, agents package and task package. All of package extensions based on the UML 2.0 core definitions. Figure 3.13 shows the core package of knowledge atlas.

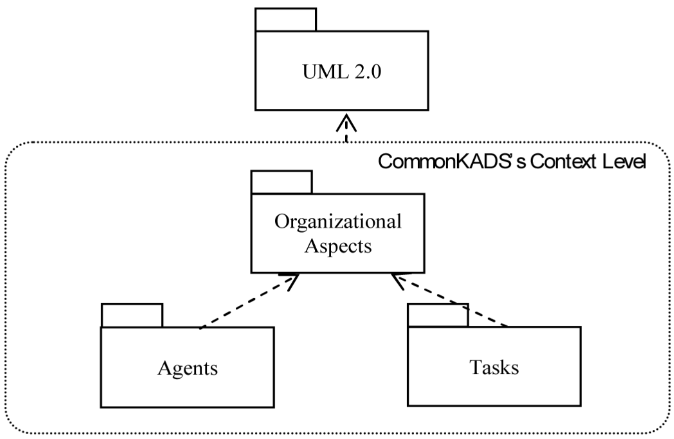


Figure 3.13: Core Package of Knowledge Atlas.

In package level, the main component is the organizational aspects package. Figure 3.13 shows the meta-class diagram that describes the knowledge atlas's architecture by using the UML extensions. From figure 3.14, the primary concepts used when defining are:

Table 3.5: Knowledge Atlas Elements and Description.

Component	Description
Structure	An organization is built from structural units.
Function	Each structural unit carries out one or more business functions.
Process	Processes describe how the work is done within the business. Processes are governed by Rules and Functions are related in time through processes.
People	People play roles in the organization. They fill positions in the structure. They sponsor certain solutions to problems and possess knowledge that is required for a function. They have responsibilities and so on.
Power	People derive power from their role in the organizational structure and from the knowledge they possess. Power plays a role in defining the problem and assessing a solution's feasibility.
Resources	The objects within the business, such as material, information, and products are used or produced in the business. The resources are arranged in structures and have relationships with each other. Resources are manipulated by used, consumed, refined, or produced through processes. Resources can be categorized into physical, abstract and informational.
Knowledge	This subcomponent is especially relevant for knowledge-based system - oriented organizational analysis. Knowledge is an organizational asset and can be described by knowledge items. A knowledge item is a collection of knowledge fragments used to perform the tasks that are defined by one or more functions. Knowledge items characterize the knowledge in the organization at a fairy general level of description, mainly for managerial purposes. The organization model also contains a list of an organization's possible knowledge bottlenecks.
Goal	The purpose of business or the outcome of business as a whole is trying to achieve. Goals can be broken down into sub-goals and allocated to individual parts of the business, such as processes of objects. Goals express the desired states of resources and are achieved by processes.

Rule	A statement defines or constraints some aspect of the business, and represents business knowledge. It governs how the business should be run or how resources may be structured and related to each other. Business rules are defined using the Object Constraint Language (OCL) which is a part of the UML standard.

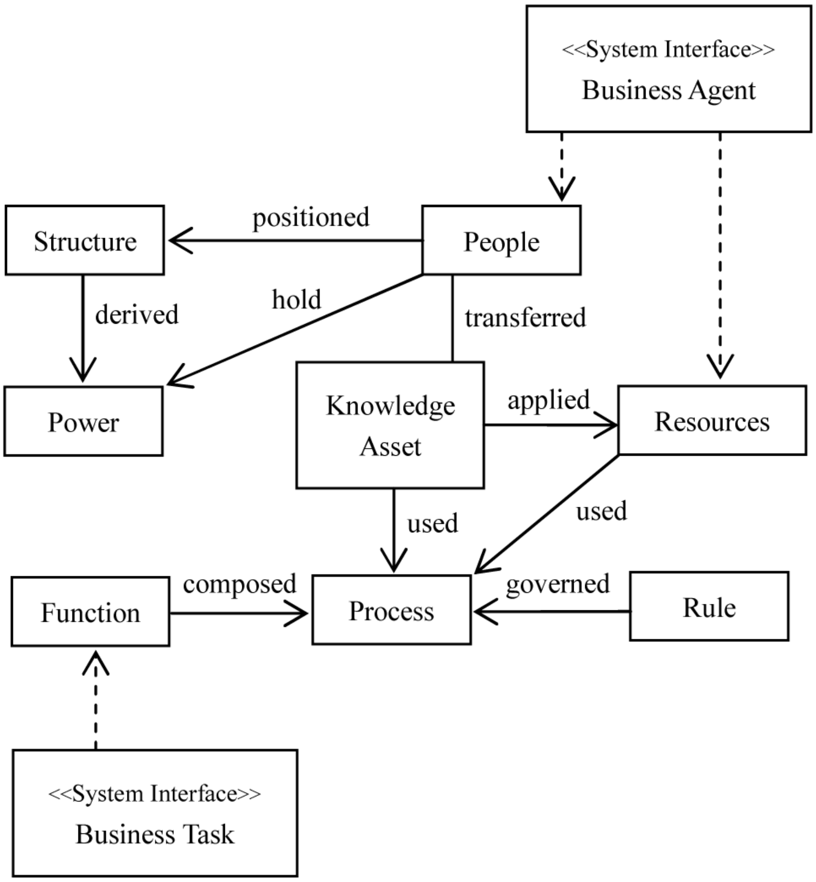


Figure 3.14: Metaclass Diagram of Knowledge Atlas with UML Extension.

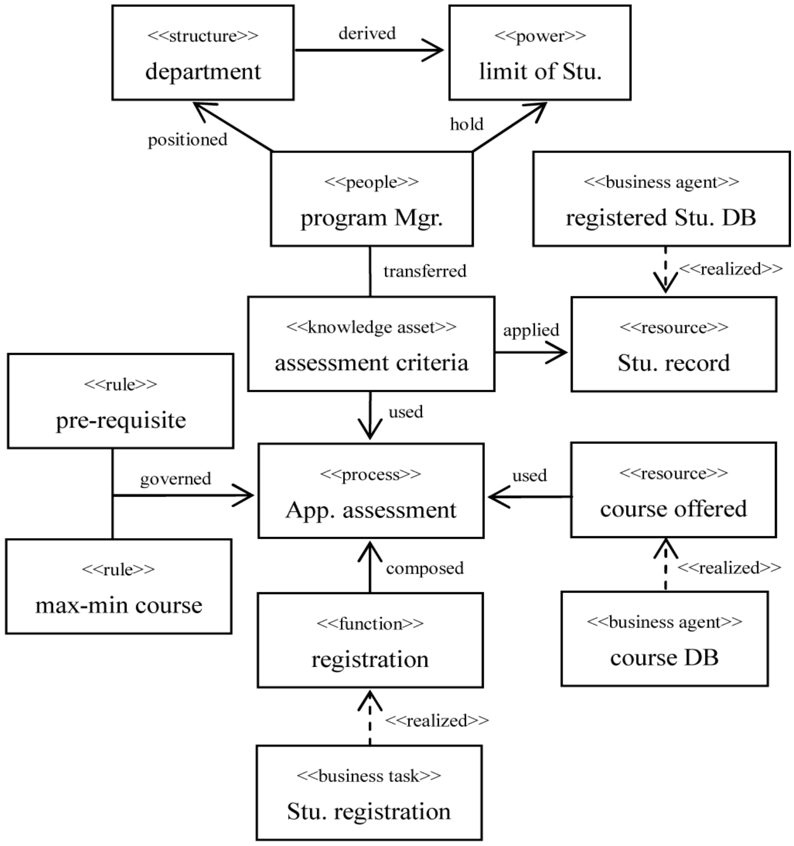


Figure 3.15: Knowledge Atlas in Online Course Registration Portal case study.

Figure 3.15 is an example schema from the case-study “online course registration portal”. In this scope, I have mainly focused concern with the “assessment criteria” for checking pre-requisites, constraints, and policies.

3.5 Knowledge Systematic Schema

This proposal is recreated from former research “A CommonKADS’s Knowledge Atlas with UML Extensions” and “CommonKADS’s Knowledge Model using Architectural View and Extension Mechanism”. The difference is former research separated virtual space development between knowledge management level and knowledge object level. This approach concluded all in one schema and one virtual space.

The main concept of elaborating knowledge in virtual space is explaining the knowledge pattern by using virtual space supportability with a manner that is understandable. The elaboration is not only to make sense of knowledge methodological, but also to support development in virtual space and design of knowledge information in XML tag-based. For these reasons, I propose the knowledge systematic schema that is recreated from CommonKADS methodological by using three architectural views: (1) physical view, (2) logical view, and (3) functional view. To realize all of views in the same virtual environment, I have developed one spot to connect those views. I call the spot as the conjugate point. The conjugate point in the schema is an abstract template class that associates role playing of knowledge relying on each selected-view. It provides template structure of three knowledge types for dynamic responsibility of knowledge.

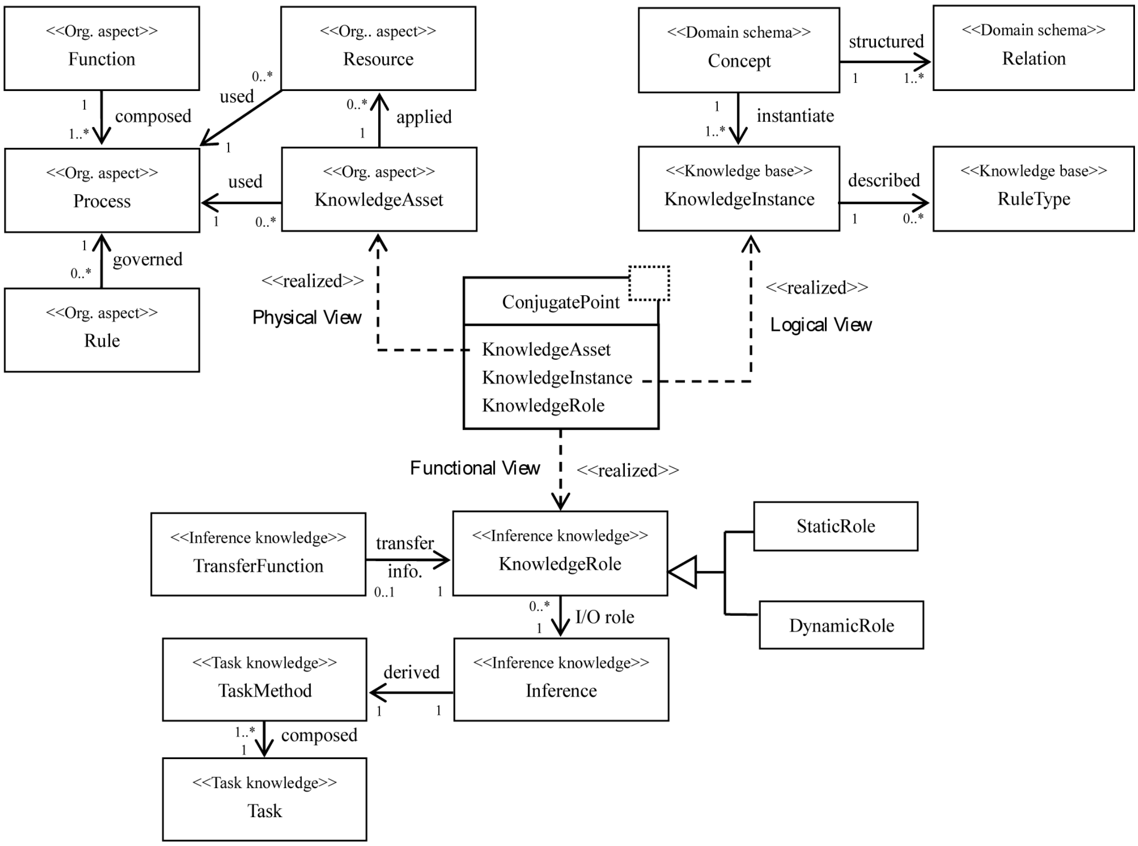


Figure 3.16: Architectural View and Knowledge Systematic Schema.

Physical view realizes the role of conjugate class as the knowledge asset class in organization aspect. Logical view realizes the role of knowledge instance class in knowledge base. Functional acts as the knowledge role class of inference structure. Figure 3.16 shows the architectural view and metadata diagram of schema.

About the view concept, Physical view represents an organization aspect that contributes to the understanding of knowledge context. It acts as an infrastructure for facilitating the interoperability of geographically-distributed development.

Logical view represents an abstraction of domain knowledge and classifies the finding knowledge with concept and their relation. It acts as the moderator of real world and abstract space of knowledge between knowledge asset in physical view and knowledge role in functional view via knowledge instance. Additionally, logical view encourages the quantization mechanism of knowledge for the correlated abstraction of knowledge instance and concept.

Functional view realizes the scenario from the knowledge intensive task that corresponds with inference and task knowledge. Its purpose is to develop a new idea by determining a chronicle order to inference knowledge as knowledge scenario and performing by touring timeline service. Additionally, this view is used to predict an expired knowledge and trace-back the expired-chain for updating the knowledge base in logical view.

3.6 Comparison of CommonKADS Model and Proposed Model

This section shows a comparison of CommonKADS model that consist of knowledge model and organization model to compare with the proposed model that consists of knowledge landscape schema, knowledge atlas schema, and knowledge systematic schema.

Table 3.6 Comparison of CommonKADS Model and Purposed Method

Technology features	CommonKADS Approach		Proposed Approach		
	Knowledge Model	***Organization Model***	***Landscape Schema***	***Atlas Schema***	***Systematic Schema***
Knowledge Engineering Technology	✓	✓	✓	✓	✓
Artifacts	2 diagrams (Domain K. diagram, Inference structure)	5 tables (OM.1-5)	1 diagram (Meta diagram)	1 diagram (Meta diagram)	1 diagram (Meta diagram)
Platform Independent	✓	✓	✓	✓	✓
View / Architectural view	2 views (Domain K Diagram, Inference structure)	1 view (Organization aspect)	2 views (Logical view, Functional view)	1 view (Physical view)	3 views (Physical view, Logical view, Functional view)
Object-oriented approach	Only Domain knowledge	-	✓	✓	✓
Standard modeling language	-	-	✓	✓	✓
Documentation	✓	✓	✓	✓	✓
Consistency checking	-	-	✓	-	✓
Extensibility	-	-	✓	✓	✓
Exchanging	-	-	✓	✓	✓

3.7 Chapter Conclusion

Knowledge model is an important part of the CommonKADS methodology and general knowledge based system. In spite of the process for constructing knowledge model is similar to other software system but it has not standard technique available for developing. UML is a general purpose modeling language that covers a wide range of different application domain. It is a standard modeling and could be adopted for the knowledge model development with its feature. This thesis chapter describes an UML approach for developing the knowledge model in CommonKADS methodology by using the architectural view and extension mechanisms feature. This approach provides model perspectives and extensible notations for modeling the knowledge model in the same context within standard of UML. The model consists of two levels: architectural level and metaclass (component) level. In methodology, I used two model perspectives for develop in an architectural level: logical view and functional view. Logical view supports view of abstract representation for knowledge type definition as objects. Functional view supports view of scenario that correspond with the usage of inference and task knowledge for define the interactive objects. All defined objects will be correlated by relationship in metaclass diagram of the component level. To define the metaclass, I used the extension mechanism such as stereotype for define knowledge model's concept. The result from methodology is the knowledge model that developed within CommonKADS concept and relies on the UML standardization.

In this chapter, I propose three schemas for develop the prototype system: (1) knowledge landscape schema, (2) knowledge atlas schema, and (3) knowledge systematic schema. Knowledge landscape schema is representative schema for the knowledge model in content level of CommonKADS. Knowledge atlas schema represents an organizational aspect in context level. Both of schemas are concluded in one schema for implementation in virtual environment called, knowledge systematic schema.

Reference Publication:

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : "CommonKADS's Knowledge Modeling using UML Architectural View and Extension Mechanism", The 7th Intl. Conf. on Advanced Information Management and Service, Jeju, Korea (ICIPM 2011).

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : "A CommonKDAS's Knowledge Atlas with UML Extensions", The Joint International Symposium on Natural Language Processing and Agriculture Ontology, Bangkok, Thailand (SNLP 2012).

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : "Conjugate of Knowledge Items Between Abstract and Organization Knowledge Models", The 4th International Conference on Digital Information and Communication Technology and Its Applications, Bangkok, Thailand (DICTAP 2014).

Bibliography

[1] Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga, "*Knowledge Engineering and Management: The CommonKADS Methodology*", A Bradford Book, The MIT Press (1999).

[2] Speel, P., Schreiber, A. Th., Van Joolingen, W., and Beijer, G., "*Conceptual Models for Knowledge-Based Systems, in Encyclopedia of Computer Science and Technology*", Marcel Dekker Inc., New York (2001).

[3] Mohd Syazwan Abdullah, Chris Kimble, Ian Benest, and Richard Paige, "*Knowledge-based systems: a re-evaluation*", Journal of Knowledge Management, vol. 10, no.3, pp.127-142 (2006).

[4] Maarten Sierhuis, "*An Object-Oriented Design Method for Knowledge Based Systems*", Proceeding of IEEE/ACM International Conference: Developing and Managing Expert System Program, pp.296-303 (1991).

[5] Mohd Syazwan Abdullah, Andy Evans, Ian Benest, Richard Paige, and Chris Kimble, "*Knowledge Modelling Using the UML Profile*", Proceedings of the 3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI), (June 2006), Athens, Greece, IFIP International Federation for Information Processing, pp.70-77 (2006).

[6] Guus Schreiber, Bob Wielinga, and Robert de Hoog, "*CommonKADS: A Comprehensive Methodology for KBS Development*", IEEE Intelligent Systems, vol. 9, pp.28-37, December (1994).

[7] M.S. Abdullah, I.Benest, A.Evans, and C.Kimble, "*Knowledge Modelling Techniques for Developing Knowledge Management Systems*" 3rd European Conference on Knowledge Management, Dublin, Ireland, ISBN:0-9540488-6-5, pp.15-25, September (2002).

[8] Gruber, T.R., "*Toward Principles for the Design of Ontologies Used for Knowledge Sharing*", Stanford University (1993).

[9] Gomez-Perez, A., Benjamins, R.R., "*Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*", IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm Sweden (1999).

[10] Mohd Syazwan Abdullah, Andy Evans, Ian Benest, Richard Paige, and Chris Kimble, "*Modelling Knowledge Based Systems Using The eXxcutable Modelling Framework (XMF)*", Proceedings of the 2004 IEEE Concerfence on Cybernetics and Intelligent Systems, pp.1055-1060, December (2004).

[11] Devedzic, V., "*Knowledge Modeling – State of the Art*", Integrated Computer-Aided Engineering, vol. 8, no. 3, pp.257-281 (2001).

[12] Garden, N. and Garden, Y., "*An Application of Knowledge Based Modeling using Scripts*", Expert System with Applications, vol.25, no.4, pp.555-568 (2003).

[13] Hakansson, A., "*A UML as an Approach to Modeling Knowledge in Rule-based Systems*", 21st SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence (ES2001), Peterhouse College, Cambridge, UK (2001).

[14] Cuena, J. and M. Molina, "*The Role of Knowledge Modeling Techniques in Software Development: a General Approach Based on a Knowledge Management Tool*", International Journal of Human-Computer Studies, vol.52, pp.385-421 (2000).

[15] Wilfried Post, Bob Wielinga, Robert de Hoog, and Guus Schreiber, "*Organizational Modeling in CommonKADS: The Emergency Medical Service*" IEEE Expert: Intelligent Systems and Their Applicationd, vol. 12, pp. 46–55, November (1997).

[16] Guus Schreiber, Bob Wielinga, and Robert de Hoog, "*CommonKADS: A Comprehensive Methodology for KBS Development*", IEEE Intelligent Systems, vol. 9, pp.28-37, December (1994).

[17] Paul D. Scott and Robert C. Vogt, “*Knowledge Oriented Learning*”, Proceeding of eight International of joint Conference on Artificial Intelligence IJCAI’83, vol.1, pp.432-435 (1983).

[18] Perez-Martinez, J.E., “*Heavyweight Extensions to the UML Metamodel to Describe the C3 Architectural Style*”, ACM SIGSOFT Software Engineering Notes, vol.28, no.3 (2003).

[19] Vladimir Sklenar, Martin Radvansky, and Michal Dobes, "*Navigation in Knowledge-Based System for Helpdesk Based on FCA*", ICCS 2007, LNAI 4640, pp. 501-504 (2007).

[20] Kende, R., “*Knowledge Modeling in Support of Knowledge Management*”, Lecture Notes in Artificial Intelligence, 2070, pp.107-112 (2001).

Chapter 4

Knowledge Realization

This chapter describes about how to realize knowledge in virtual space and prototype system. The realization is based on problem and opportunities and research goal relationships as following in table 4.1:

Table 4.1: Traceability of Problem and Opportunities and Research Goal with Related Realization.

Problems and Opportunities*	Research Goals*	Realization
[P-1]	[G-1]	*Knowledge and virtual space design*
	[G-2]	*Realized mechanisms*
	[G-3]	*Knowledge and virtual space design and Realized mechanisms*
[P-2]	[G-4]	*Knowledge and virtual space design and Realized mechanisms*
	[G-5]	*System specification and architecture*
[P-3]	[G-2]	*Realized mechanisms*
	[G-3]	*Knowledge and virtual space design and Realized mechanisms*
	[G-5]	*System specification and architecture*
[P-4]	[G-2]	*Realized mechanisms*
	[G-3]	*Knowledge and virtual space design and Realized mechanisms*

** reference number from research purpose in chapter 1, page 12.*

From the traceability table the first column is the problems and opportunities which define in research purpose in chapter 1. The second column is research goals which response to the problem and opportunities. For example, problem [P-1]: "Organization knowledge is a key asset in an organization but it is often tacit and private. From the survey, most systems lack of explain how an organization uses it knowledge is built up". The responsive of [P-1] in research goals are: [G-1]: "Aim to develop useful and practical guidelines for knowledge intensive organization by develop the schema is to get acquainted with the system and to assess the amount of foreknowledge needed", [G-2]: "Enables one to spot the opportunities and bottlenecks in how organizations develop, distribute and apply their knowledge resources, and so gives tools for corporate knowledge management", and [G-3]: "Provide the methods to obtain a thorough understanding of the structures and processes used by knowledge workers even where much of their knowledge is tacit leading to a better integration of information technology in support of knowledge work". The last column is how to answer the research goal with realization in this chapter.

For the description: Section 4.1 describes knowledge and virtual space design that consists of three subsections: scene-graph design for virtual space in 4.1.1, knowledge representation in 4.1.2, and simulation scene of knowledge space in 4.1.3. Section 4.2 describes about realized mechanisms: topological of knowledge in 4.2.1 and knowledge scenario development life cycle in 4.2.2. Section 4.3 is system description and its architecture. Section 4.4 is system features and users interfaces. Finally, section 4.5 is chapter conclusion.

4.1 Knowledge and Virtual Space Design

To develop a knowledge methodology in virtual space, I develop from the knowledge systematic schema in chapter 3 section 3.5. It aims to explicate in detail of type and structure of knowledge used in performing task and also concentrate on conceptual structure of knowledge. I used this schema to design a scene-graph for an imaginary guideline before implementation. The benefit from manage knowledge in virtual space are simplicity and well-known by learning from experiences in controllable spiral that understandable in human-sense. It is an important vehicle role for communication between experts and users during both developing through system execution.

4.1.1 Scene-graph Design for Virtual Space

Knowledge schema has purpose to explicate in detail of types and knowledge structure which used in performing task. Although the prototype system demonstrates with Google™ API which have feature functions for managing scene-graph by itself, but some API functions, developer need the concept of scene-graph for insight implementation. To describe scene-graph concept, I use concept of the WorldToolKit™ (WTK) for demonstration. This concept is a generality which can apply for other programming languages, such as, Java 3D API, etc.

Generally, a common definition of a graph is a data structure composed of nodes and arcs. A node is data element, and arc is relationship between data elements. To render scene-graph, WTK provides function for creating nodes and placing them at specific positions in the scene-graph. Developer might be concerned the scene-graph on the common rules as followed:

- The scene-graph is rendered automatically into the window as the simulation runs.
- Different scene-graphs may have common sub-trees. This means that the same geometry can be referenced by more than one scene-graph.
- Each scene-graph has a single root node.
- Traversal begins at the root node of scene-graph.
- The renderer traverses the tree from top to bottom and left to right.

Depending on the type of node, WTK will do different things. Nodes in WTK can be grouped into three distinct types: Geometry node, Attribute node, and Procedural node.

Geometry nodes contain the representation of visible entities by draw the specified set of polygons. Attribute nodes used to affect the way geometry nodes are rendered by modify the current state, which determines an appearance of subsequent geometry. Procedural nodes used to control the way a scene-graph is put together by process the children of this node, depending on the type of traversal directed by the node. For managing the state of the scene-graph, Separator and Transform Separator nodes are used to manage state of scene-graph by isolating the effects of attribute nodes.

From knowledge schema, I design a general scene-graph shown in figure 4.1 (left). Scene-graph starts at the root node. The root node has relationships with three nodes. Light node specifies lighting of the landscape on the virtual space. Transform node sets a position and orientation information for develop Task method into Task knowledge. The transform separator manages the sub-graph of Inference knowledge level. On Inference level, it consists of geometry node of Inference knowledge, Transfer function, and Knowledge role. The transform separator in this level separates Domain knowledge level by scope of the knowledge base. In this level, it has concept node and level of detail (LOD) node that describes knowledge instance and attribute instance in geometry node. For figure 4.1 (right) shows a scene-graph for Google™ earth. The different is scene-graph for Google™ Earth do not need configure environment.

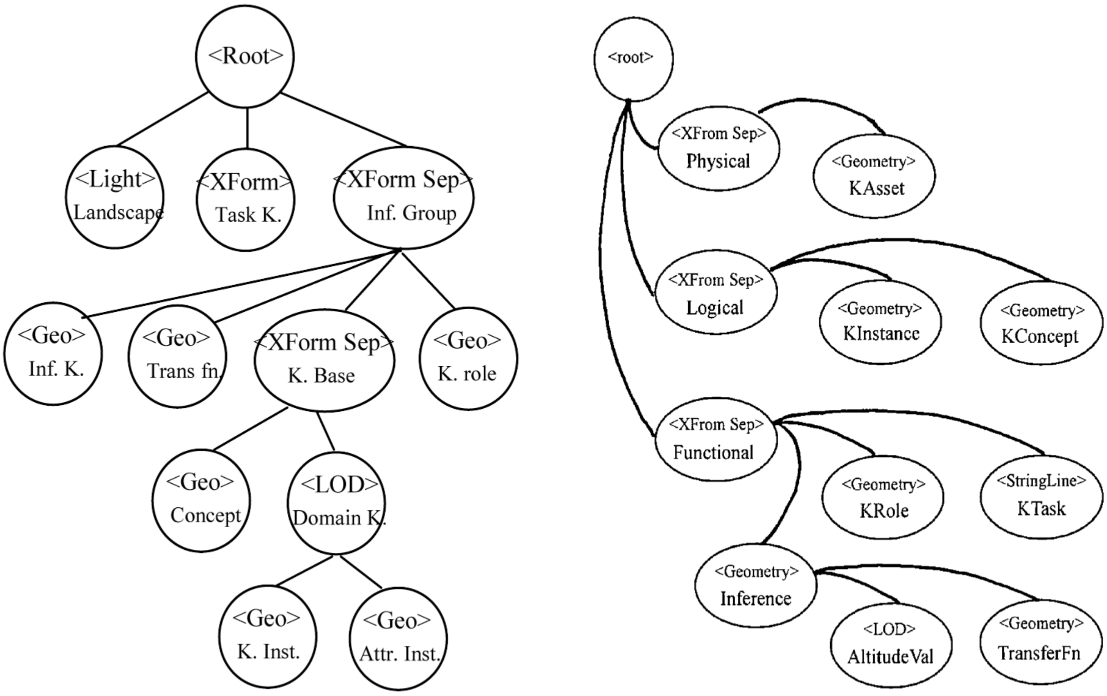


Figure 4.1: Knowledge in the virtual space scene-graph.
(left): Scene-graph for WTK, (right): Scene-graph for Google™ Earth.

4.1.2 Knowledge Representation

For added acquired knowledge to system, knowledge might be transformed into the knowledge item that represents with XML-formatting file. It is composed of meta-knowledge and hyperlink of information body. Meta-knowledge is abstract of information body and hyperlink of information body obtains the detailed content of knowledge. Users can access to information body through hyperlink, so as to implement the integration of operations.

In knowledge structure, it consists of three tag parts: card name, description and hyperlink of information body. Figure 4.2 shows a sample of knowledge in an XML file. Considered tag of XML file, the <card> element represents a unit of a knowledge item. It contains three child elements: <name>, <description>, and <bodylink>. The <name> element contains a name of knowledge card that impile to knowledge item, the <description> elements contains the short abstraction of knowledge item, and the <bodylink> element contains the URL of an embedded file (e.g., an image, a movie clip, a slide, and so on).

```
<?xml version="1.0" encoding="UTF-8"?>
<card version="1.0">
<name>   Checked BIOS </name>
<description>
Check BIOS setting that compatibility with memory
</description>
<bodylink> IT_SUPPORT/BIOS/chkBIOS.mpg </bodylink>
</card>
```

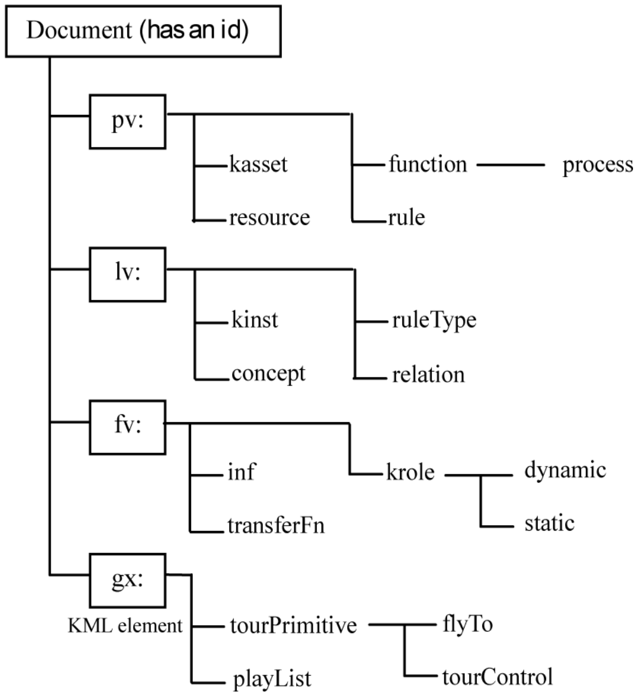


Figure 4.2: Diagram of KML/XML-tag.

For knowledge specification, I use an approach of middle-in and middle-out techniques to specify the knowledge. The middle-out approach is preferred, but can only be used if the inference structure of the task template is already at the required level of detail. If decomposition is necessary, the process essentially becomes “middle-in”. Deciding on the suitability of the inference structure is therefore an important decision criterion. Figure 4.3 shows middle-in and middle-out approaches for knowledge specification.

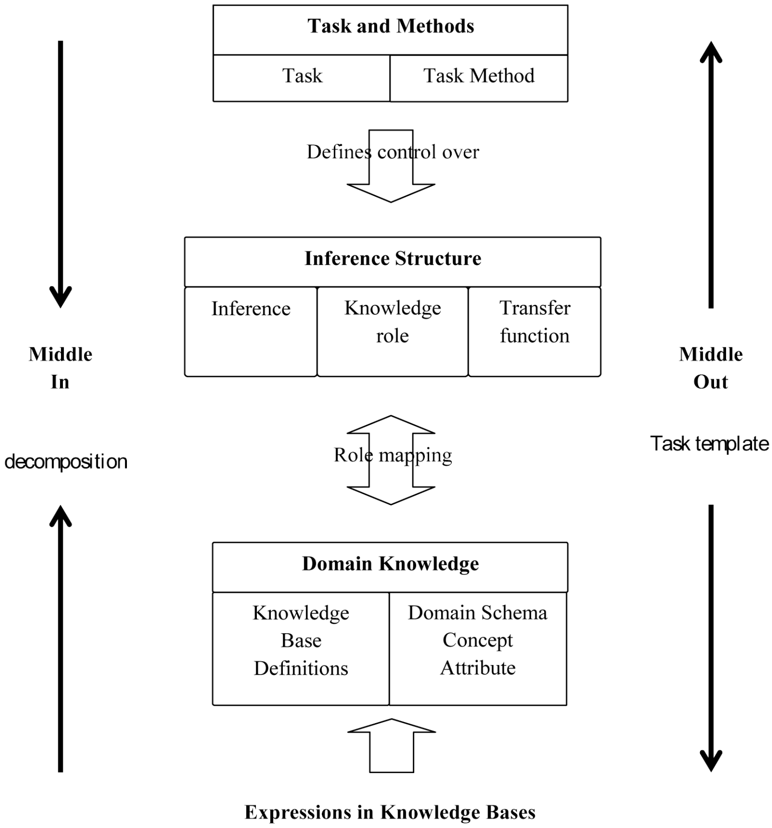


Figure 4.3: Middle-in and Middle-out Approaches to Knowledge Specification.

4.1.3 Simulation Scene of Knowledge Space

The concern of knowledge realization in virtual space is how difficult to learn to work and how to clarify required of the underlying knowledge representation. First of all, I designed a scenegraph to realize schemas that represent knowledge elements with geometry-node, an application user can visually grasp the global nature of node, explore the information space, and accommodate new at an appropriate place based on manner understandable by the abilities of virtual space. I considered develop the Google™ APIs interface both of Map and Earth for implement the virtual space prototype, on reasons to investigate the macroscopic view of knowledge and the participating geographically distributed development. About the element description, I explain via KML/XML tag-based schemas, relied on the schema definition and use meta-knowledge for access information body via the hyperlink, so as to implement an integrated knowledge operation. The level of detail (LOD) depends on the zooming interface value and priority of node type, for example, inference node zooming priority more than transfer function node, etc. User can use the camera control for getting current view, searching the node, panning, tiling, etc. based on the geocoding application features. To realize schemas in virtual space, I separate the virtual space by geographical altitude in three layers: physical layer, logical layer, and functional layer.

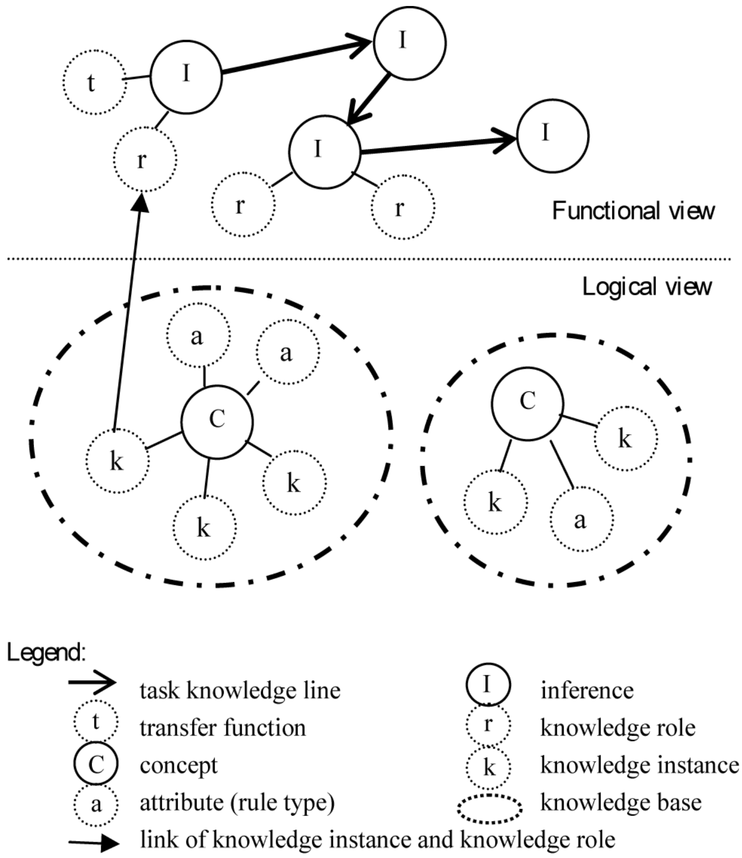


Figure 4.4: Simulation-scene Diagram of Virtual Space Design.

Physical layer is a ground layer that represents knowledge assets in organization and conducted by Knowledge Atlas schema. User can develop the knowledge asset via node <placemark> and display <balloon> for detail, such as holder, resources, process, etc. Additionally, they can customize an icon for more insight dimension, for example, using man-icon for the tacit knowledge and book-icon for the explicit knowledge, etc.

Logical layer is a middle layer, located between physical and functional layer. It contains an abstraction of domain knowledge that conducted by the logical view of Knowledge Landscape schema. It acts as the moderator between real-world and abstract-world of knowledge by linking between knowledge asset on physical layer and knowledge role in functional layer via by knowledge instance. Furthermore, its view encourages the mechanism of knowledge quantization on the correlated abstraction among knowledge instance, concept, and their relation.

Finally, functional layer is an upper layer that conducted by functional view of Knowledge Landscape schema. Its purpose to develop the new idea by determines a chronicle order of inference knowledge, as knowledge scenarios for the system perform by touring timeline. Additionally, I use this layer to predict the expired knowledge and trace-back the expired-chain for updating the knowledge base. Figure 4.4 shows the scene of the virtual space in 2D diagram.

4.2 Realized Mechanisms

4.2.1 Topological of Knowledge

In virtual knowledge memory space, it comprised of knowledge node that has own latitude, longitude, and zooming value for determine user's viewpoint. The geographical value of node enables user to judge the location reply on knowledge schemas and spatial clues. Figure 4.5 shows the topological of knowledge node and filtration developing.

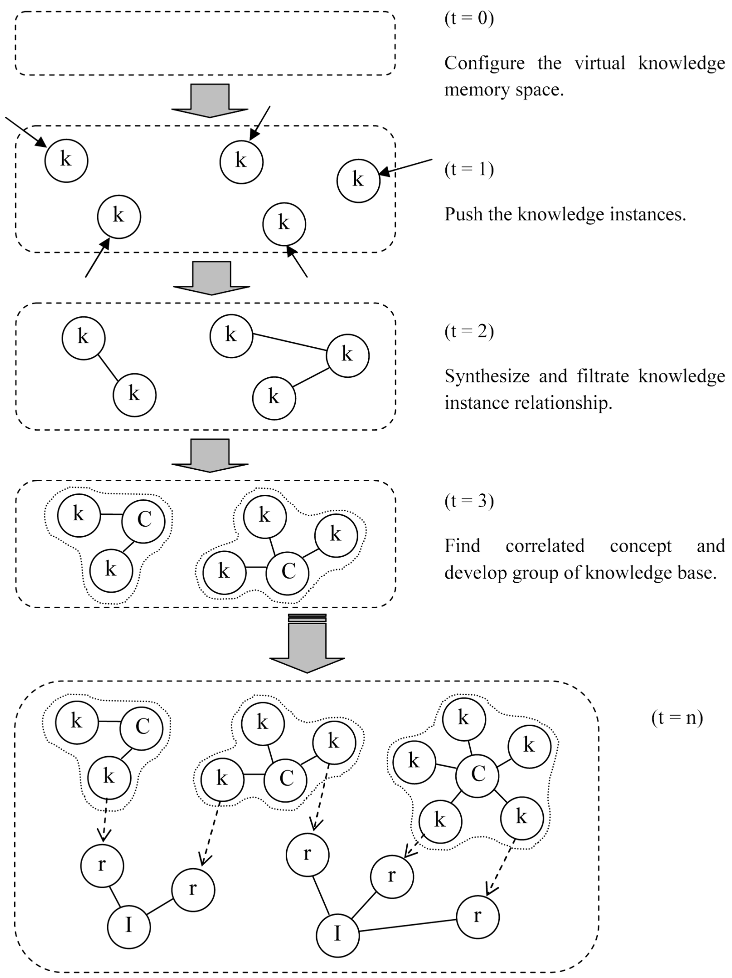


Figure 4.5: The Topological of Knowledge and Filtration Developing.

4.2.2 Knowledge Scenario Development Lifecycle

Knowledge scenario is the sequencing related to step of inference knowledge usage. It similarly likes functional decomposition and method in computer programming. Typically, each of knowledge scenarios has only one individual goal to achieve as atomicity goal.

To develop the knowledge scenario, user might determine the chronicle order to inference knowledge element, such as, inference and/or transfer function. The chosen inference knowledge display its ordering number and the linking line to the others on a step, as a task knowledge. The structure and length of scenario depend on the situation occurs and the solution to achieve a goal. User can edit knowledge scenario by rearranging or combining old and new the inference knowledge as life cycle. Figure 4.6 shows the knowledge scenario developing: scenario (A) and scenario (B) and their reusability.

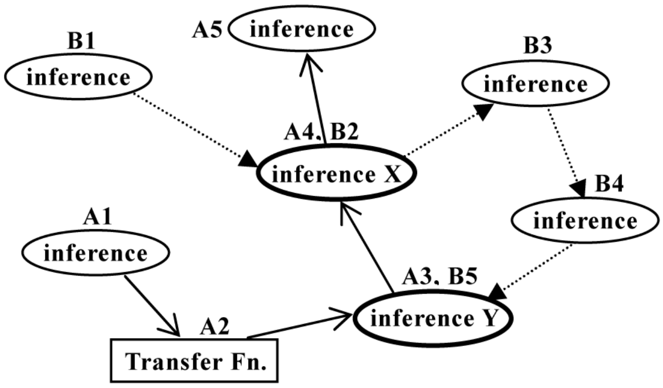


Figure 4.6: The Knowledge Scenario (A) and (B) and Reusability of Inference X and Y.

From figure 4.6, knowledge scenario (A) represents with thick-line has 5 steps (a1) - (a5). knowledge scenario (B) represents with dot-line has 5 steps (b1) - (b5). Both of knowledge scenarios have two shared 2 inferences: inference X and inference Y.

Example

This example describes two task knowledge have an aim to write a basic 3D program. Task (a) is written by Java^{TM} 3D and Task (b) is written by WorldToolKit^{TM}.

Table 4.2: Example of Knowledge Scenario Development: Reusable and Shareable.

Task (a): Java 3D^{TM}	**Task (b): WorldToolKit^{TM}**
(a1) Create a Canvas3D object. (a2) Create a VirtualUniverse object. (a3) Create a Locale object, attaching it to the VirtualUniverse object. (a4) Construct a view branch graph. (a4.1) Create a View object, ViewPlatform, PhysicalBody, and PhysicalEnvironment object. (a4.2) Attach Canvas3D, ViewPlatform, PhysicalBody, and PhysicalEnvironment object to View object. (a5) Construct content branch graph(s). (a6) Complie branch graph(s).	(b1) Create a WTuniverse. (b2) Entered simulation by calling WTuniverse_go. (b3) Read sensors. (b4) Call Universe action function. (b5) Perform object tasks. (b6) Play/Record paths. (b7) Render the Universe.

From two above tasks, the sequencing number of step (a1) - (a6) and b(1) - b(7) are the task method that similar to knowledge scenario. Each of steps in task method equal as an inference. Some of inferences from task (a) and (b) could be shared, for example, inference (a2) and (b1). Because of these inferences contain the Knowledge instances that have same Concept such as, VirtualUniverse and WTuniverse are the Knowledge instance that have the Concept of creating of virtual area.

4.3 System Specification and Architecture

For system specification, I separated the specification with two dimensions: Developing tools specification in table 4.3 and Environment configuration in table 4.4.

Table 4.3: Developing Tools Specification.

Developing Tool	Specification
Google™ Code Playground	Modifying API Coding AJAX Unit Testing
EditPlus 3.51	Coding AJAX and JavaScript Develop XML (Knowledge Schema) Develop KML + XML
IBM Rational Rose, Microsoft Visual C++	Design Schema Develop UML Model (e.g,Use-Case)
WorldToolKit™ 9, Java 3D	Design Scenegraph Concept

For environment configuration, the application is developed on the system platform below:

Operation system#1: Windows 7 Home Premium SP1
Operation system#2: Windows XP SP3
CPU#1: Intel Core-i5
CPU#2: Intel Core-i3
RAM: 4 GB
Browser: Internet Explorer 10.0.10

For more information about the Google™ plugin, table 4.4 is currently supported on the following platforms:

Table 4.4: Google™ Plugin Supporting Platform.

Microsoft Windows	Apple Mac OS X 10.5 and higher (Intel)
Google Chrome 5.0+ Internet Explorer 7.0+ Firefox 3.0+ Flock 1.0+	Google Chrome 5.0+ Safari 3.1+ Firefox 3.0+

To install the Google™ plugin, browser to any webpage in which the plugin is included (e.g. http://code.google.com/apis/ajax/playground/#hello,_earth). The browser will likely ask for permission before installing plugin – click through to allow installation. Once the plugin is installed, user may need to refresh the page before the plugin display correctly.

For the application software architecture, I designed the system with software layer architecture. It consists of four layers: Application layer, Business layer, Middle-ware layer, and Physical layer. For the application layer, it composes of web browser as an application interface and Google™ Map / Earth plugin for interpret KML to render on the virtual space. On the business layer, it composes of a module for compile the XML that represent the knowledge information rely on the knowledge schemas, both of Knowledge Landscape and Knowledge Atlas. The module is developed with AJAX for compile the XML file by construct the object tree model (DOM – Data Object Model). For using the knowledge schema via XML, it is used together with the KML-tag by encapsulation the knowledge schema-XML with KML-tag. On the middle-ware layer, it provides the Google™ API and KML namespace as the fundamental infrastructure. Finally the physical layer, it focuses on the communication for interoperability of the system via the internet. Figure 4.7 shows the software architecture with the layer architecture.

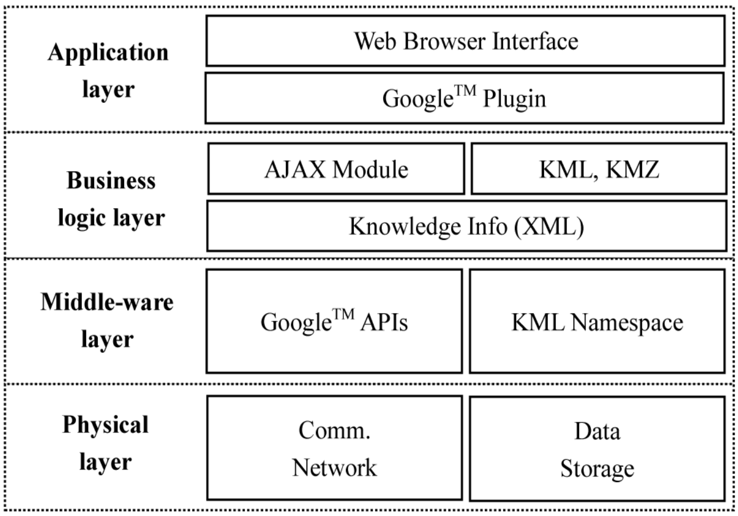


Figure 4.7: Architectural Layer (Software Architecture).

For the system architecture, I elaborated the architecture via the Web-base Client-Server architecture, due to the system features use the Google™ Map and Earth API from the Google™ server-side. For the knowledge information, it is stored in two ways: the first way is deploy the database on the internet / WAN for globalized sharing. The second way is deploy the database in an organization via intranet network. The different of two options is the usability and security. However, knowledge engineer can use both two ways by separated the critical knowledge into the intranet scope. For the prototype system, I developed by sharing text file. The text file is lightweight sharing in case the number of knowledge is not large. For import and export knowledge information, the system provides an interface for transform the information via standard-XML to/from the system. Figure 4.8 shows the system tier architecture.

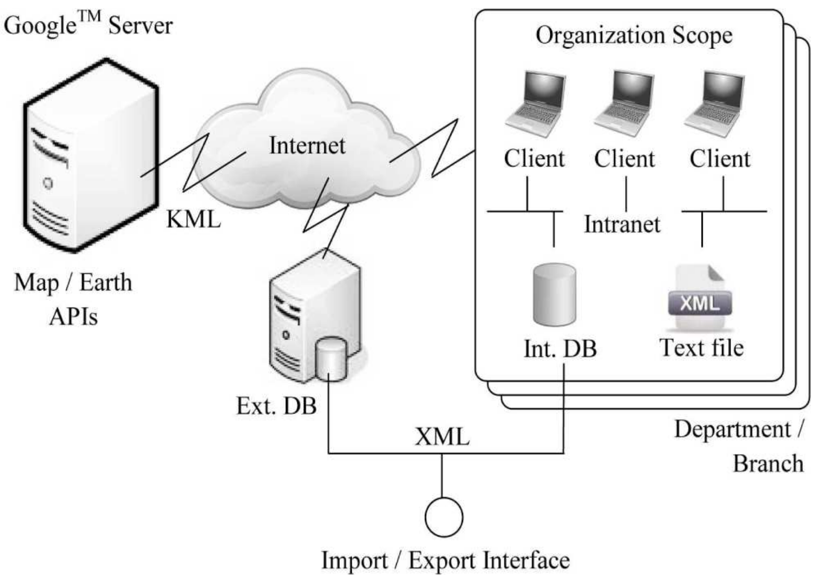


Figure 4.8: Architectural Tier (System Architecture).

4.4 Features and User Interface

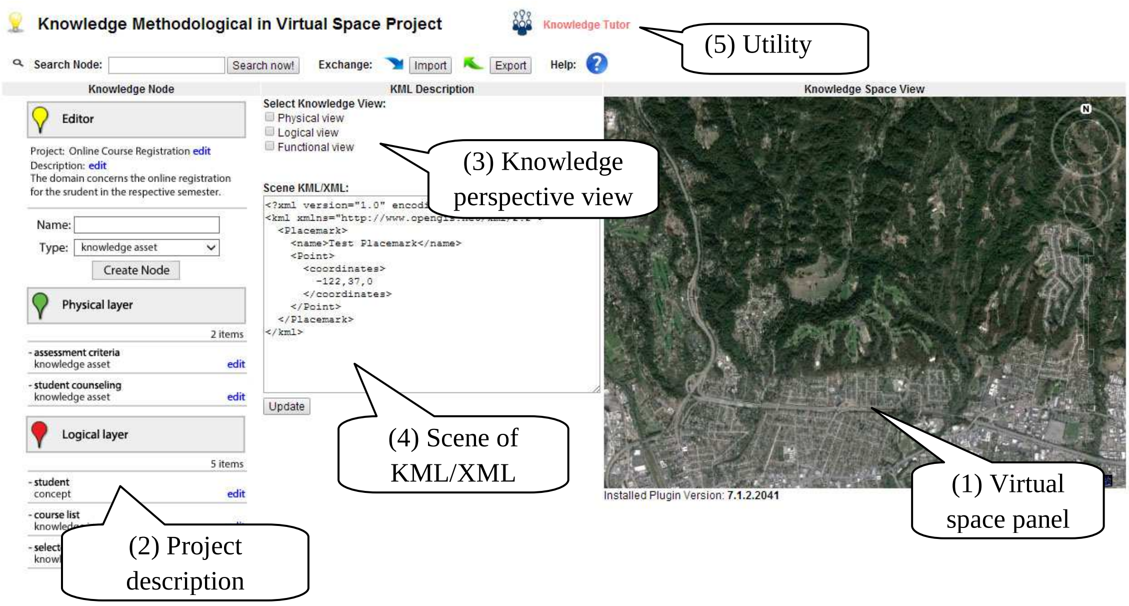


Figure 4.9: Prototype User Interface.

Figure 4.9 is the prototype system interface that consists of 5 panels: (1) virtual space panel, (2) project description panel, (3) selected knowledge view, (4) KML/XML scene for editor, and (5) utility panel.

Virtual space panel is an operational knowledge virtual space function; user can create, update, and manipulate knowledge node directly with virtual space features. All of information on virtual space panel is linked to project description panel via KML/XML data. User can select node in project description panel for update information detail in case of user non-familiar operating on virtual space, and for observe knowledge hierarchy.

For knowledge perspective view, user can select and hide perspective view for inspection knowledge layer on virtual space. KML/XML scene for editor shows data of selected node in project description with KML/XML-format for advanced user directly editing, as shown in figure 4.10.

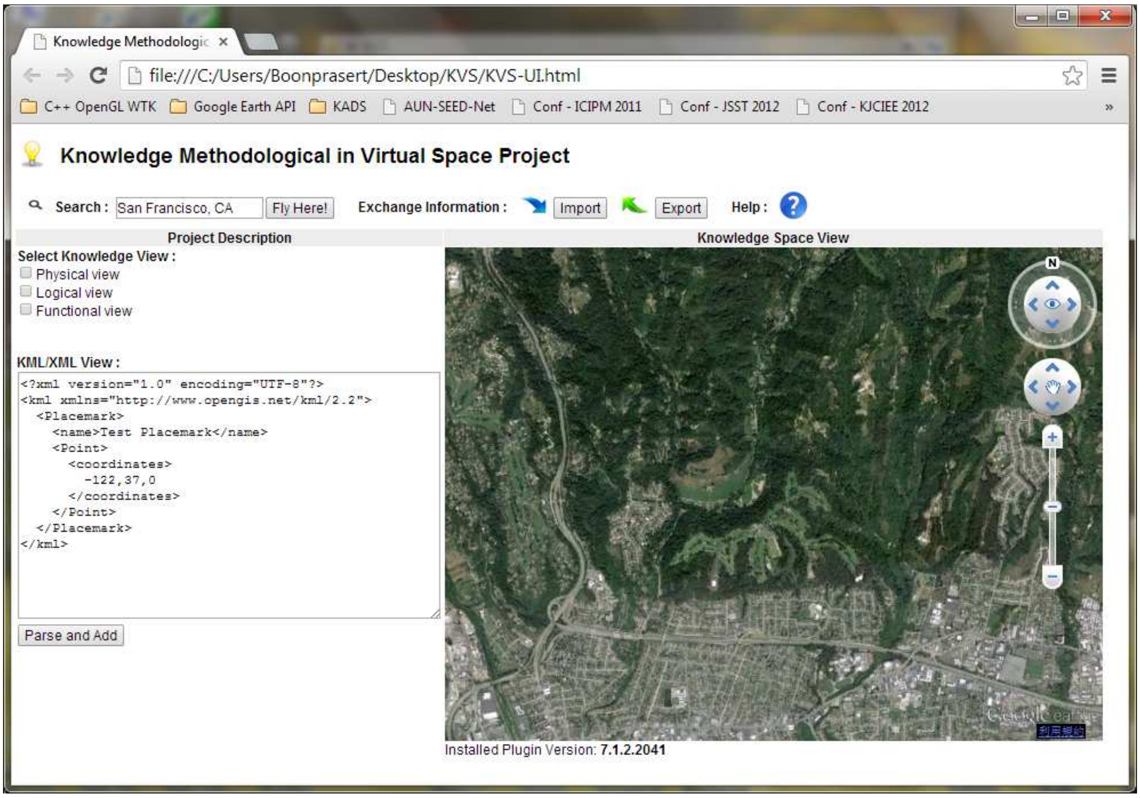


Figure 4.10: Interface for KML/XML Investigation.

On utility panel, it consists of the peripheral tools such as, searching node, import and export information, help information, and knowledge tutor for assistance naïve user.

4.5 Chapter Conclusion

Managing knowledge through knowledge memory system is an important part of knowledge management initiative. I use the CommonKADS methodology concept to be the guideline for develop the schemas of knowledge content and organization context for display in the virtual space. In virtual space, I represent knowledge item by node. Each of knowledge node composed of latitude, longitude and zooming values that determine the user field of vision. To delivery knowledge, user creates the knowledge scenario by the series of knowledge node, called knowledge scenario. The advantage for define knowledge in a scenario are reusability and modifiability. User can create a new knowledge scenario with previous knowledge and filtrate knowledge forward to the sustainable knowledge memory system.

For knowledge realization, I develop the knowledge memory space on the virtual space, it is a memory system that has an objective view enables user to edit contents on surface by using geographical arrangement and topological connection. It provides an overview of large data contents and facilitate for knowledge sharing on people and increasing their connectivity as a lightweight activity.

I propose a scene-graph to implement the knowledge schema regimen. The scene-graph is not only explicating graphical simulation, but also has control over the decision of server-side and client-side rendering to encourage the groupware system. To describe the elements of knowledge, I encapsulate knowledge informatics in XML with Keyhole Markup Language (KML). The system extracts the information using DOM-parser and manipulates knowledge information with AJAX implemented module. Finally, I demonstrate the proposed approach by prototyping a system developed in Google^{TM} Earth APIs environment as virtual environment.

From an elaborate of knowledge realization responsive research goal in this chapter, the problems and opportunities issues have been solved as following:

[Problem-1]: *Organization knowledge is a key asset in an organization but it is often tacit and private. From the survey, most systems lack of explain how an organization uses it knowledge is built up.*

This problem was solved by developing of knowledge systematic schema which compounded from both of concept and context level in knowledge discipline, especially knowledge atlas schema from context level that indicate an organization aspect and focus on organization knowledge development. Finally, knowledge systematic schema was realized in virtual environment by scene-graph and knowledge representation with its mechanisms.

[Problem-2]: *From the survey, many systems lack of an interface understandable manner and suggested usability in user perform.*

This problem was solved with creation of a virtual environment that developed for knowledge-developing process. The proposed system used virtual technology to simulate abstract space and real-world space of knowledge developing in virtual environment, with an objective to connect abstract space and real world space in an understandable manner. An user interface of virtual environment is designed base on how difficult is it to learn to work and how to clarify required of the underlying knowledge representation.

***[Problem-3]:** From the survey, some system has not methodological support and lack of collaborative work to improve knowledge exchange.*

This problem was solved from knowledge schemas that developed and modified by take an advantage from CommonKADS methodology by optimized two models: knowledge model and organization model. Finally, the proposed schema was realized in virtual environment that developed in web-based application and architecture for sharing and exchange knowledge information enhancement via network communication.

***[Problem-4]:** From the comparison, some technique has complicated models and most of frameworks are non-standardization language for knowledge-developing process.*

In this study, I solved the complicated and various models by using architectural model view for given multi-perspective in each of model, and concluded all in one model for realization in a virtual environment. For the standardization, I customized the UML language with extensions mechanisms to define knowledge elements and regiment.

Reference Publication:

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : "Scene-Graph Design for Knowledge Memory Space", The 31st International Conference on Simulation Technology, Kobe Japan (JSST 2012).

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : "Knowledge Schema for Knowledge's Virtual Space", Kansai Joint Conference of Institute of Electronic and Engineering (KJCIEE 2012).

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : "Knowledge Memory System Design on Virtual Space", Proceeding of ITE Tokai University, Vol.5, No.1, pp.13-19 (2012).

Bibliography

[1] Kubota H., Nomura S., Sumi Y., and Nishida T., "*Sustainable Memory System using Global and Conical Spaces*", Journal of Universal Computer Science, vol.13 no.2, pp.135-148 (2007).

[2] Bally J.M., "*Designing Workspace: an Interdiscipline Experience*", SIGCHI Conference on Human Factors in Computing Systems, pp.10-15 (1994).

[3] Toyoaki Nishida, "*Conversational Informatics an Engineering Approach*", John Wiley & Son Ltd. England (2007).

[4] Minoh M., and Nishiguchi S., "*Environment Media – In the case of Lecture of Archiving System*", KES2003, vol. II, pp.1070-1076 (2003).

[5] Bederson B.B., and Hollan J.D., "*Padd++: A Zooming Graphical Interface for Exploring Alternate Interface Physics*", UIST'94, pp.17-26 (1994).

[6] Dennis J Bouvier, "*Getting Started with the Java 3D API*", Sun Microsystems (1999).

[7] Leno Franzen, Hans Kessock, and Dave Hinkle, "*WorldToolKit Reference Manual*", Engineering Animation inc., April (1999).

[8] Ryosuke Saga, Akinori Kageyama, and Hiroshi Tsuji : "*Proposal for Re-Usable TODO Knowledge Management System RESTER*", IEEJ Transactions on Electronics, Information and Systems, vol.126, no.4 (2006).

[9] Hiroyuki Kojima, and Ken Iwata : "*Seamless Management of Paper and Electronic Documents for Task Knowledge Sharing*", IEEJ Transactions on Electronics, Information and Systems, vol.130, no.4 (2010).

[10] Studer R., Benjamins V.R., Fensel D.: "*Knowledge Engineering Principles and Methods*", Data and Knowledge Engineering, vol.25, pp.161-197 (1998).

[11] Andriessen JHE : "*Working with Groupware: Understanding and Evaluating Collaboration Technology*", Springer, London (2003).

[12] Kouji Kozaki, Yoshinobu Kitamura, Mitsuru Ikeda, Riichiro Mizoguchi : "*Hozo : An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship*", Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), pp.213-218 (2002).

Chapter 5

Evaluation and Discussion

This chapter describes about the experiment strategies that designed based on research goal as following in table 5.1:

Table 5.1: Traceability of Research Goal and Experiment Strategies.

Research Goals*	**Experiments**	**Evaluation Strategies**
[G-1]	[Exp.1]	*Feature comparison*
	[Exp.2]	*Questionnaire: basic features*
[G-2]	[Exp.2]	*Questionnaire: basic features*
		Questionnaire: collaborative work
[G-3]	[Exp.2]	*Questionnaire: environment driven suggestion*
		Questionnaire: collaborative work
		Questionnaire: collaborative work
[G-4]	[Exp.2]	*Questionnaire: environment driven suggestion*
	[Exp.3]	*Process of experimental task*
[G-5]	[Exp.1]	*Feature comparison*
	[Exp.3]	*Process of experimental task*

** reference number from research purpose chapter 1, page 12.*

From the table 5.1, the first column is the research goals that refer from research purpose in chapter 1. Second column are experiments: [Exp.1] is feature comparison experiment, [Exp.2] is user questionnaire response, and [Exp.3] is the process of experimental task. The last column is evaluation strategies, especially in experiment 2, have 3 types of questionnaire: about environment-driven suggestion, about basic features, and about collaborative work supportability.

This chapter consists of two sections. First section 5.1 describes about the evaluation strategies: feature comparison in 5.1.1, user questionnaire response evaluation in 5.1.2, and process of experimental task in 5.1.3. Section 5.2 is the research discussion.

5.1 Evaluation

In this research, both qualitative and quantitative evaluations were employed in this study. A comprehensive evaluation of knowledge intensive organization schema is the major focus is put on static characteristics of the proposed system. Evaluation concentrated on characteristics of the knowledge-development process supported by methodology. I provided three strategies for evaluations: features comparison, questionnaire response, and process of experimental tasks. The set of criteria that will be used for comparing divided into the following groups:

General description which includes information about developers, release and availability.

Software architecture and tool evolution which includes information about the tool architecture (standalone, client/server, n-tier application), how the tool can be extend with other functionalities/modules, how information are stored (database, text files, etc.) and if there is any backup management system.

Interoperability with other tools and language which includes information about interoperability capacities of the tools. I will review the tool's interoperability with other ontology tools (for merge, annotation, storage, inferencing, etc.), as well as translations to and from ontology language.

Knowledge representation I will present KR paradigm underlying the knowledge model of the tool. It is very relevant in order to know what and how knowledge can be modeled in the tool.

Usability I analyze the existing of graphical editors for the creation of concept taxonomies and relations, the ability to prune the graphs and the possibility to perform zooms of parts of it. I will also analyze if the tool allows some kind of collaborative working.

5.1.1 Experiment 1: Features Comparison

To evaluate the different engineering tools, I specified a number of relevant criteria in three dimensions. First there is a general dimension, which refers to aspects of the system that can also found in other types of programs.

The second dimension refers to information about the knowledge-developing supportability and different actions the user can perform. Relevant questions would include: Meaning of interface is easy to support step of work? The clarity by interface is clear? Interface enable judging facts based on internal parameter? Is there a good overview and particular view? Does the system check new data for consistency? The various concepts in this system were well integrated? etc.

The last dimension is that of interoperability, which is used to evaluate the tool's support for constructing by several people at different locations. For examples of questions: Does the system allow synchronous editing by different users? Provided features are enough for the needs of the corporation? Is it possible to import information from another tool? Is it possible to export information in various format? etc.

I evaluated the system features by comparing with survey of the other knowledge tools, such as, Protégé 2000 and WebODE. Seven knowledge domain users participated in the study; male 2 persons and female 5 persons; age between 25-36 years with age average 28.57. All participants had experience in knowledge and/or ontology tools more than 1 year with experience average 2 years and 4 months.

The comparison consists of two sections: general description survey and comparative questionnaire. For questionnaire, I complied into a 3-level scale (+, 0, -) subsequently calculating a weight mean of the results. A plus (+) means positive, e.g. the feature/characteristic is available or properly implemented. A zero (0) means reasonable, e.g. the feature is available, but it is difficult to use. A minus (-) is negative, e.g. the feature is not supported or not correctly implemented. "NA" means not applicable and a questionmark means that I have been unable to find out. The comparative features are divided into the followed dimensions: general description, knowledge-developing supportability, and interoperability, as shown questions in Table 5.2. Table 5.3 shows comparison of proposed system with Protégé 2000 and WebODE.

Table 5.2: The statements of comparative questions.

ID	Statements of questionnaire
q01	Provided features are enough for work
q02	Meaning of interface is easy to support step of work
q03	The clarity by interface is clear
q04	Interface enable judging facts based on internal parameters
q05	Is there a good overview and particular view
q06	Does the tool check new data for consistency
q07	The various concepts in this system were well integrated
q08	Does the tool allow synchronous editing by different users
q09	Is it possible to import an information from another tool
q10	Is it possible to export an information in various formats
q11	Provided features are enough for the needs of the corporation
q12	Are the changes made by other user easy to recognize

Table 5.3: Features comparison between the proposed system, protégé 2000, and WebODE. A plus (+) means positive, a zero (0) means reasonable, a minus (-) is negative. “NA” stands for not applicable and
a question mark means unable to find out.

Criterion	Proposed System	Protégé 2000	WebODE
General Description			
Availability	Open source	Open source	S/W license and free Web
Software Architecture	Client/Server	Standalone Client-Server	Client-Server
Extensibility	API/Plugins	Plugins	API/Plugins
Storage	File/Database	File/Database	Database
Methodological support	CommonKADS	-	Methontology
Knowledge-developing Supportability			
q01 Features supportability	+	+	+
q02 Interface-wise guidance	+	0	0
q03 Graphical taxonomy	+	+	-
q04 Graphical prunes(view)	+	+	+
q05 Zooming	+	+	-
q06 Consistency checking	+	+	+
q07 Compliance standard	0	+	+
Interoperability			
q08 Synchronous editing	0	0	-
q09 Import facilities	0	0	+
q10 Export facilities	+	0	0
q11 Collaborative working	+	0	+
q12 Change recognition	-	-	-

5.1.2 Experiment 2: Users Evaluation

The goal of this experiment was designed to evaluate supportability, usability, and utility of proposed system. The participants recruited on a volunteer basis from related knowledge domain and non-domain in King Mongkut's Institute of Technology Ladkrabang (KMITL) Thailand, for examples, staffs in office of Quality Assurance and Knowledge Management (QA&KM), researchers from knowledge laboratory, knowledge management course students, and any persons who were interest; all total 30 subjects; male 14 persons and female 16 persons; age between 19-37 years with age average of 25.13. All participants had experience in web browser such as, Internet Explorer®, and ever used application related diagram, Maps, and/or Earth browser. I provided an experimental environment with a private web server that shared KML/XML of pilot project for experiment.

In experimental process, firstly I explained the procedure of experiment, including separated all participants into three groups based on experience on related knowledge tool, such as, Protégé, Mindmap, Compendium, etc. The detail of classification as following:

- Advanced user has experience more than one knowledge tool with average 2.5, total 8 subjects, and age average 28.5.

- Experienced user has experience with one knowledge tool, total 10 subjects, and age average 20.70.

- Non-experienced/Naïve user never has experience with knowledge tool, total 12 subjects, and age average 26.58.

Table 5.4: Three participant groups and description.

User Group	Tool Exp.	Number	Sex	Age
Advanced	> 1	8	M 3, F 5	25-36
Experienced	= 1	10	M 6, F 4	19-22
Non-experienced / Naïve	0	12	M 5, F 7	22-37

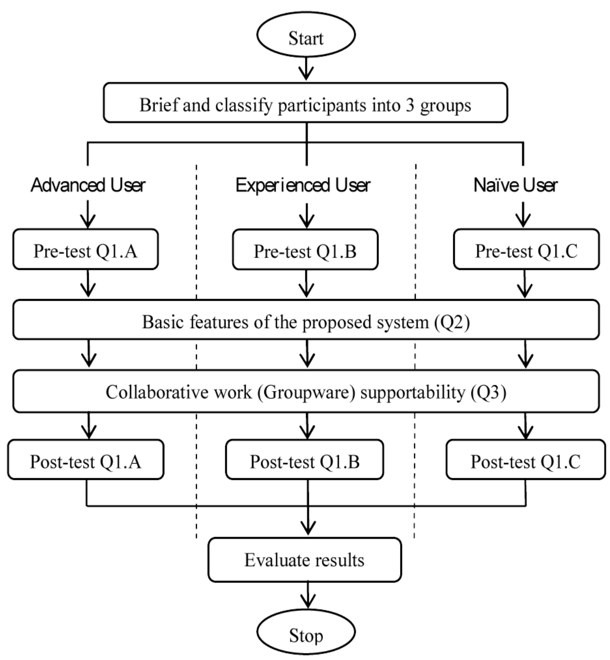


Figure 5.1: Flowchart of experimental process.

Table 5.3 shows detail of participant groups with knowledge tool experience, number of subjects, sex, and range of subject age. Figure 5.1 shows the experimental process activities.

In experimental process, subjects evaluated proposed system with three questionnaires as following:

(Q1) Pre-test and post-test of environment-driven suggestion evaluation; total 10 questions, as shown in Table 5.5.

(Q2) Basic features of proposed system evaluation; total 10 questions, as shown in Table 5.6.

(Q3) Collaborative work (Groupware) supportability evaluation; total 8 questions, as shown in Table 5.7.

For questionnaire (Q1), it was divided into three groups (Q1.A, Q1.B, and Q1.C) relies on group of participant in Table 5.4.

All of questionnaires were designed on a scale (1-5): ("1" means "Strongly disagree" to "5" means "Strongly agree"), subsequently calculating a weight mean of the results. The types of question consist of: subject perform on provided task questions, survey attitude questions, and open-ended comment and suggest. I used spreadsheet software for calculating the results. Figure 5.2 is portion of questionnaires (Q1.A) environment-driven suggestion evaluation for advanced user, and the experiment session on site, as shown in Figure 5.3.

Read the possibilities below and answer to what extent they meet your needs at work, according to the scale on the right. In this scale 1 corresponding to "it does not meet the needs" and 5 corresponds to "fully meets the needs".	
1. Evaluate the clarity of the interface	- □ □ □ □ □ + 1 2 3 4 5
2. Evaluate the consistency of the interface	- □ □ □ □ □ + 1 2 3 4 5
3. Evaluate the speed of updating after new data is inserted	- □ □ □ □ □ + 1 2 3 4 5
4. Is there a good overview of the ontology?	- □ □ □ □ □ +

Figure 5.2: Portion of questionnaire (Q1.A): environment-driven suggestion evaluation for advanced user group (experience >1 tool) on scale (1-5): 1 means "Strongly disagree" to 5 means "Strongly agree".

Figure 5.3: The experiment: *(left)* experiment in computer laboratory room and *(right)* brief and classify participants before experiment.

Table 5.5: The statements of environment-driven suggestion questions (Q1), separated on user group: (A) = Advanced, (E) = Experienced and (N) = Naïve; the (✓) is used and (×) is not used.

ID	Statements of questionnaire (Q1)	Used for		
		A	E	N
e01	Is there a good overview of the information	×	✓	✓
e02	Is there a good particle view of the information	✓	×	✓
e03	I found the interface easy for work	×	✓	✓
e04	I found the interface enough for work	✓	✓	✓
e05	I found the interface guide step of work	✓	✓	✓
e06	The meaning of information is very easy to understand	×	✓	✓
e07	The information is easy to apprehend.	✓	×	✓
e08	The information is enough for work	×	✓	✓
e09	I think the environment interface encourage my work	✓	✓	✓
e10	I think that I could contribute to this information	✓	✓	✓
e11	I found the various concepts were well integrated	✓	✓	×
e12	I imagine that most legal experts would understand this information very quickly	✓	✓	×
e13	The information enable judging adequacy of conclusion	✓	×	×
e14	The information enable judging facts based on internal parameters	✓	×	×

Table 5.6: The statements of basic features questions (Q2).

ID	Statements of questionnaire (Q2)
f01	The meaning of the interfaces are clear
f02	I am confident I understand the conceptualization of the tool
f03	I found the information very easy to understand
f04	Does the tool check new data for consistency
f05	Evaluate the speed of updating after new data is inserted
f06	Are the changes identifiable clear to user
f07	Is it possible to use multiple inheritance
f08	Is it possible to create exhaustive and/or disjoint decomposition
f09	Evaluate the stability of the tool (crashes, etc.)
f10	Are there example available in the tool

Table 5.7: The statements of collaborative work (Groupware) supportability questions (Q3).

ID	Statements of questionnaire (Q3)
g01	Does the tool allow synchronous editing by different users
g02	Provided features are enough for the needs of the corporation
g03	I found the sharing of information cumbersome to understand
g04	Are the change made by other user easy to recognize
g05	Are the ways to lock the information
g06	Is it possible to browse the information if it locked
g07	Is it possible to import an information from another tool
g08	Is it possible to export the information in various format

5.1.3 Experiment 3: Process of Experimental Task

The goal of experiment was designed to evaluate usability support of proposed system. Ten university students (are not overlap the evaluation 5.1) from knowledge management class in faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand participated in this study; male 7 persons and female 3 persons; age between 19-21 years with age average 19.9.

In experimental process, I divided participant into two groups by random 5 persons per group: (1) Training Group and (2) Non-training group. For training group, I brief how to operate on proposed system and prepared a manual in experiment. On the other hands, non-training group only observe the web browser screen before experiment 1 minute and repeat the process tasks again after all first process tasks finish.

On the process, all participants perform the process of experimental tasks in table 5.8. The provided tasks consist of three suites: (1) operate from physical to logical view, (2) operate from logical to functional view, and (3) develop scenario in function view. During operation, participant has timekeeping after each tasks suite finish. Figure 5.4 shows the experimental process activities. Table 5.8 shows the process of experimental task.

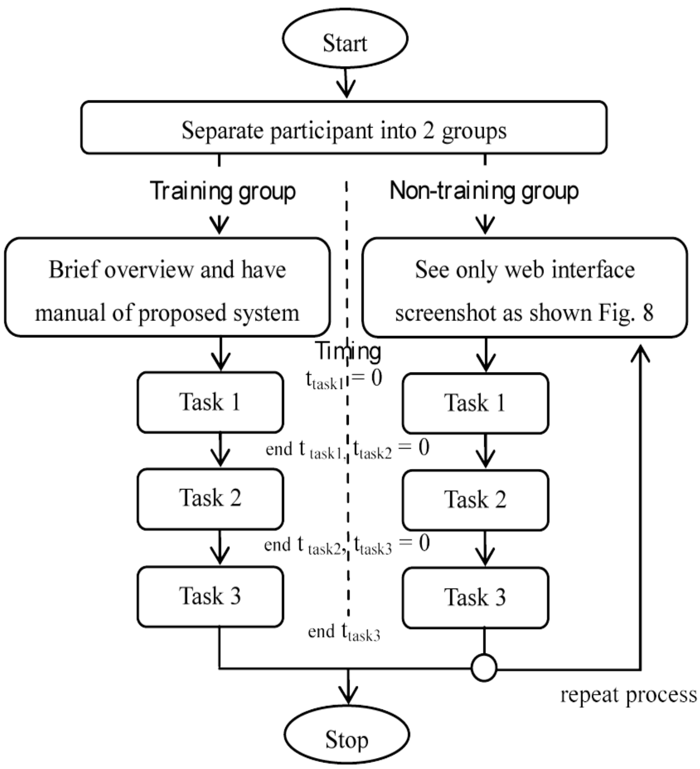


Figure 5.4: Flowchart of process experimental tasks.

Table 5.8: Process of experimental task.

Step	Task
0-0	Experiment provided KML/XML shared file and direction sheet.
1-1	Physical view: Create a Knowledge Asset, set name "KA".
1-2	Set description from direction sheet to "KA".
1-3	Switch to Logical view: Create a Knowledge Instance: "KI-1".
1-4	Set link from "KI-1" to "KA".
2-1	Logical view: Create a Concept: "C" and Knowledge Instance: "KI-2".
2-2	Set link from "KI-1" and "KI-2" to "C".
2-3	Switch to Functional view: Create 2 Knowledge Roles: "KR-1" and "KR-2".
2-4	Set link "KI-1" to "KR-1" and "KI-2" to "KR-2".
3-1	Functional view: Create Inference: "I-1" and set description from sheet.
3-2	Set link "KR-1" to "I-1" and Transfer function: "TF-1" to "I-1".
3-3	Create touring scenario: "I-1" to "I-2" to "I-3" and display touring.
0-0	End of experiment.

5.2 Discussion

Advantages of the proposed system can be observed from features comparison in Table 5.3 Compared with other knowledge tools, our system provides advance usability that supports by virtual space interface for knowledge development process. The strength point in knowledge-developing supportability is the interface-wise guidance. Because of the proposed system is designed based on start up at organization context, so user can develop bottom up approach with their environment. On the other hand, Protégé 2000 and WebODE start up at conceptual knowledge. For interoperability, because of this research was developed using virtual geographically environment, so it can enhance collaborative work.

From the results of experiment 5.1.2 shown in Table 5.9 and Figure 5.5 suggest that:

- Average means of post-test are greater than pre-test: participants of all levels repeated the same task with better understanding by environment-driven interface supportability.

- Difference value between pre-test and post-test of each group: $\Delta d(Q1.B) = 0.92 > \Delta d(Q1.C) = 0.90 > \Delta d(Q1.A) = 0.56$: an impact of environment-driven gives more clues to beginner than advanced user.

Table 5.9: Result of average means and standard deviation in (Q1) pre-test and post-test of environment-driven suggestion of three user groups: Advanced, Experienced, and Naïve.

User Group	Pre-test		Post-test	
	Mean	SD	Mean	SD
Advanced	2.80	± 0.25	3.36	± 0.27
Experienced	2.70	± 0.33	3.62	± 0.23
Naïve	2.43	± 0.22	3.33	± 0.28

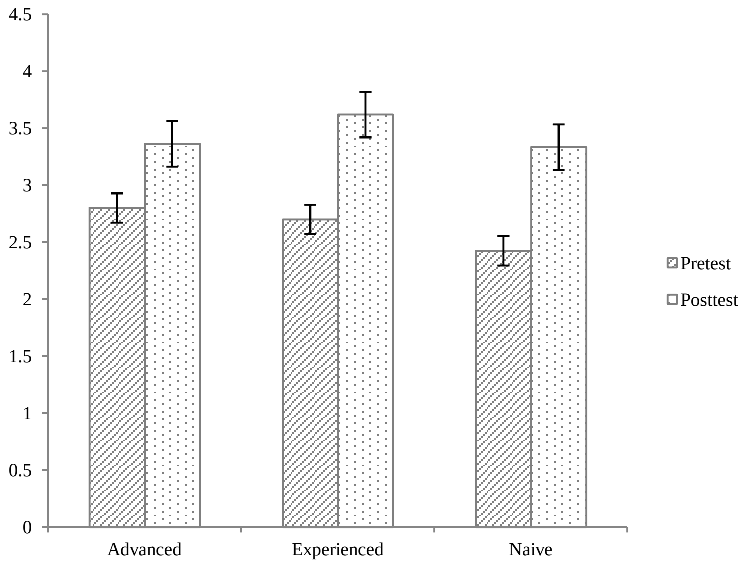


Figure 5.5: Comparison of average between pre-test and post-test of environment-driven suggestion questionnaire (Q1) separated in three participant groups. The participants were differentiated by experience on knowledge tool: Advanced (tool exp. >1), Experienced (tool exp. =1), and Naïve (never used).

To test the earlier stated hypothesis, t-test is performed at 95% confidence level, with following results in Table 5.10.

Table 5.10: t-test summary in (Q1) pre-test and post-test of environment-driven suggestion of three user groups: Advanced, Experienced, and Naïve.

Group	**Paired Differences**		**t**	**Df**	**Sig. 2-tailed**
	Mean	Std. error			
Advanced	5.63	0.53	10.57	7	0.000
Experienced	9.20	0.59	15.53	9	0.000
Naïve	9.08	0.19	47.07	11	0.000

As such, based on the above t-test results, post-test in overall is significantly better than pre-test in Q1. All participants are better understanding by environment-driven interface supportability, especially beginner user.

From the evaluation results in (Q2) and (Q3) shown in Table 5.11 and Figure 5.6, the average means of (Q2) Basic features and (Q3) Collaborative work (groupware) supportability are high. This results show that the proposed system satisfies the wider user with various experiences.

Table 5.11: Result of average means and standard deviation in Basic features (Q2) and Groupware supportability (Q3) questionnaires.

User Group	**Basic features (Q2)**		**Groupware (Q3)**	
	Mean	SD	Mean	SD
Advanced	3.55	± 0.66	3.20	± 0.63
Experienced	3.56	± 0.46	3.09	± 0.57
Non-experienced / Naïve	3.48	± 0.38	3.02	± 0.54
Total	3.53	± 0.50	3.10	± 0.58

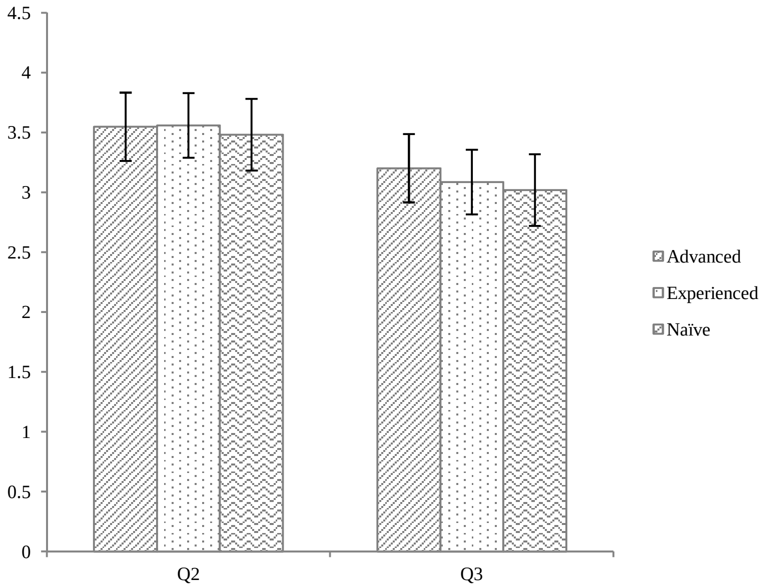


Figure 5.6: Comparison of average in (Q2) basic features and (Q3) Collaborative work (groupware) supportability questionnaires. The participants were differentiated by experience on knowledge tool: Advanced (exp. tool >1), Experienced (exp. Tool =1), and Naïve (never used).

For the open-ended comments, through analyzing their answer, I not only attempt to understand their perspective, but also to utilize these important suggestions for future improvements. I concluded in Table 5.12 as following:

Table 5.12: Responses of experimental participants.

Participants	**Suggestions**
PP01, PP05	An interface is designed for supporting the macroscopic view. It is suitable for the system that has location and area is significant factor.
PP05, PP11	Some features use altitude value to define the node. It is not familiar for non-experienced user.
PP08	The proposed system is simple and can be developed rapidly. It is suitable for using knowledge in ad-hoc situation, such as, in case of disaster, emergency planning, and basic tool for knowledge management class.
PP12	The proposed system is easy to use, supports wide range of users, and provides better understanding for knowledge in context.
PP24, PP27	The proposed system could not support in case of knowledge structure more complicated and have big data.

The result of experiment 5.3 shown in Table 5.13 presents a total of time cost on three tasks, compared between training group and non-training both the first-time and the second-time experiment. This result can be shown that learning does not take too much time, so that can easily learn in proposed system. The second time usage of non-training group less than the first time usage. This result can be shown user can learn by themselves without material suggestion and learning with experience from environment suggestion satisfies same as material guideline.

Table 5.13: Result of average means and standard deviation
in process of experiment task of training user group and non-training user group with spending time (unit: seconds).

Task	**Training Group**		**Non-training Group**			
			First		**Second**	
	Mean	SD	Mean	SD	Mean	SD
Task 1	76.6	± 8.20	110.2	± 11.84	74.6	± 8.93
Task 2	131.2	± 12.68	174.8	± 24.41	147.8	± 17.33
Task 3	176.6	± 15.79	203.2	± 28.89	182.2	± 12.0

To test the earlier stated hypothesis, t-test is performed at 95% confidence level, with following results in Table 5.14.

Table 5.14: t-test summary in process of experiment task of training user group and non-training user group with spending time.
TR = Training group; NTR#1 = Non-training, the first time process;
NTR#2 = Non-training, the second time process.

Task	**Compared between** (group #time)		**Paired Differences**		**T**	**Sig. 2-tailed**
			Mean	Std. error		
1	TR	NTR#1	33.60	4.73	7.11	0.002
	TR	NTR#2	2.00	5.97	0.34	0.744
2	TR	NTR#1	43.60	15.31	2.85	0.046
	TR	NTR#2	16.60	13.34	1.24	0.281
3	TR	NTR#1	26.60	14.54	1.83	0.141
	TR	NTR#2	5.60	10.46	0.54	0.621

df = 4

Based on above t-test results, the total of time cost on task 1 and task 2 are significantly self-learning, by time cost of non-training group on the second time are less than the first time. On the other hand, task 3 is non-significantly, though it has same direction as task 1 and task 2.

5.3 Chapter Conclusion

From an evaluation responsive research goal in this chapter, the research goals have been achieved as following:

[Goal-1]: *Aim to develop useful and practical guidelines for knowledge intensive organization by develop the schema is to get acquainted with the system and to assess the amount of foreknowledge needed.*

In an experiment 1, feature comparison shown the result of proposed system has enough features for work with others tools in the current market. Moreover, it was developed based on knowledge schema compounded from an organization aspect. Additionally in experiment 2, the results of questions in questionnaire about information need are good in practical view, enough for work, and easy to understand.

[Goal-2]: *Enables one to spot the opportunities and bottlenecks in how organizations develop, distribute and apply their knowledge resources, and so gives tools for corporate knowledge management.*

This goal was achieved by an experiment 2. The results from questionnaire suite had shown the proposed system provided features for knowledge-developing process rely on a knowledge systematic schema that developed from organization aspect, and encourage collaborative work and distribute knowledge for corporate knowledge management.

[Goal-3]: *Provide the methods to obtain a thorough understanding of the structures and processes used by knowledge workers even where much of their knowledge is tacit leading to a better integration of information technology in support of knowledge work.*

In an experiment 2 provided the pre-test and post-test to evaluate environment-driven suggestion supportability. The significant of difference between pre-test and post-test shown user could understand the structure and process knowledge with their experience to perform the system. The result of basic features is high shown the proposed system provided enough function for knowledge developing. Additionally, an average of collaborative work supportability is high, shown the proposed system has base of knowledge integration fundamental.

[Goal-4]: *Designed concern how difficult is it to learn to work with the system and about the amount of knowledge required of the underlying knowledge representation language.*

This goal was achieved with evaluate experiment 2 and 3. For an experiment 2, the result from questionnaire shown the user satisfies provided features and environment-driven supportability with high level. It provided an interface that easy to work, guide step of work, and encourage user working. On the other hand, an experiment 3 shown the non-training user group could learn to work on the proposed system by themselves without material guideline. Additionally, they could learn by their experience as well as training user group.

[Goal-5]: *Build better knowledge system that easier to use, has a well-structured architecture, and simpler to maintain.*

This goal was investigated in an experiment 1 and experiment 3. In an experiment 1, the provided features both of knowledge-developing supportability and interoperability are enough for work and satisfy by compare with other systems. The stability of system architecture was developed base on well-known stable architecture in web-base application, both of software and system design. For experiment 3, the result of self-learning in non-training user group shown the system is easier for work. They could learn with experience without material guideline and not much learning time.

Reference Publication:

Boonprasert Surakratanasakul and Kazuhiko Hamamoto : “An Approach to Design A Virtual Space to Support Knowledge Methodological Environment”, The Transactions of the Institute of Electronic Engineers of Japan (IEEJ), Section.C, Vol.134, No.12. (in publishing, 2014.10.11)

Bibliography

[1] Riichiro Mizoguchi, and Kouji Kozaki : “Ontology Engineering Environments”, Handbook on Ontologies, International Handbooks on Information Systems, pp.315-336 (2009).

[2] Duinevel A., Weiden M., Kenepa B., and Benjamis R. : “*WonderTools? A Comparative Study of Ontological Engineering Tools*”, Proceeding of KAW99, Banff (1999).

[3] Andriessen JHE : “*Working with Groupware: Understanding and Evaluating Collaboration Technology*”, Springer, London (2003).

[4] Kouji Kozaki, Yoshinobu Kitamura, Mitsuru Ikeda, Riichiro Mizoguchi : “*Hozo : An Environment for Building/Using Ontologies Based on a Fundamental Consideration of “Role” and “Relationship*””, Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), pp.213-218 (2002).

[5] Madokoro T., Nakatsuji M., and Okamoto K., Miyazaki S., Harada T. : “*Know-who/know-how Navigation Using Development Project-related Taxonomies*”, IEEE International Conference on Semantic Computing, pp.559-560 (2009).

[6] York Sure, Jurgen Ahqele, and Steffen Staab : “*OntoEidit: Guiding Ontology Development by Methodology and Inferencing*”, Proceeding On the Move to Meaningful Internet Systems, ISBN: 3-540-00106-9, pp.1205-1222 (2002).

[7] N.F. Noy, M. Sintek, S. Decker, M. Crubezy, R.W. Fergerson, and M.A. Musen : “*Creating Semantic Wen Contents with Protégé-2000*”, IEEE Intelligent Systems vol.16, no.2, pp.60-71 (2001).

[8] Gomez-Perez A., Angele J., Fernandez-Lop ez M., Christophides V., Stutt A., Sure Y., et al. : “*A Survey on Ontology Tools. OntoWeb Deliverable 1.3*”, Universidad Politecnia de Madrid (2002).

Chapter 6

Conclusions

Our economic and social life is becoming more knowledge-driven. The need of tool supports the knowledge is required. The tool should include ready to use and collaborative supportability that covers wider users with various experiences in web-based information system. In this study, I present a novel knowledge intensive organization model in virtual environment based on CommonKADS methodology. The proposed system was developed by using knowledge systematic schemas covering knowledge management level and knowledge object level. I used the scene-graph for construction and explained systematic description with KML/XML-based, additionally demonstrated the proposed approach by prototyping a system developed in Google™ Earth APIs environment.

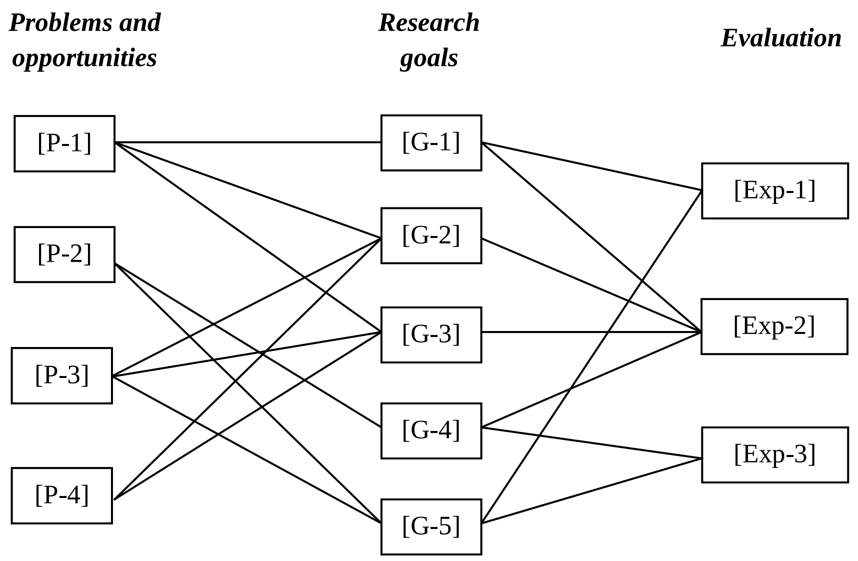


Figure 6.1: Thesis Traceability.

Figure 6.1 shown thesis traceability from problems and opportunities to research goals and evaluated experiments. The line is link between traceable elements for ensure that all elements have been developed. For the reference number on elements refer from: problems and opportunities [P-1] – [P-4] in Chapter 1, section 1.3.1, research goals [G-1] – [G-5] in Chapter 1, section 1.3.2, and Evaluation [Exp-1] – [Exp-3] in Chapter 5. Section 5.1.

In this study, I purpose a new knowledge intensive model with three schemas: knowledge landscape, knowledge atlas, and knowledge systematic schema. (1) Knowledge landscape schema elaborates knowledge model concept with logical view and functional view. (2) Knowledge atlas realizes an organization aspect for understanding knowledge environment. (3) Knowledge systematic schema is an approach concluded all in one schema with three views in one environment, and using conjugate class for associate knowledge role playing on each selected-view. Finally, I demonstrated the prototype application that developed with knowledge systematic schema in virtual environment. The results of experiment show that the proposed system improves knowledge methodology in various experience user levels for supportability, usability, and utility. Additionally, its convergent design improves knowledge methodological suggestion for wider user with various experiences.

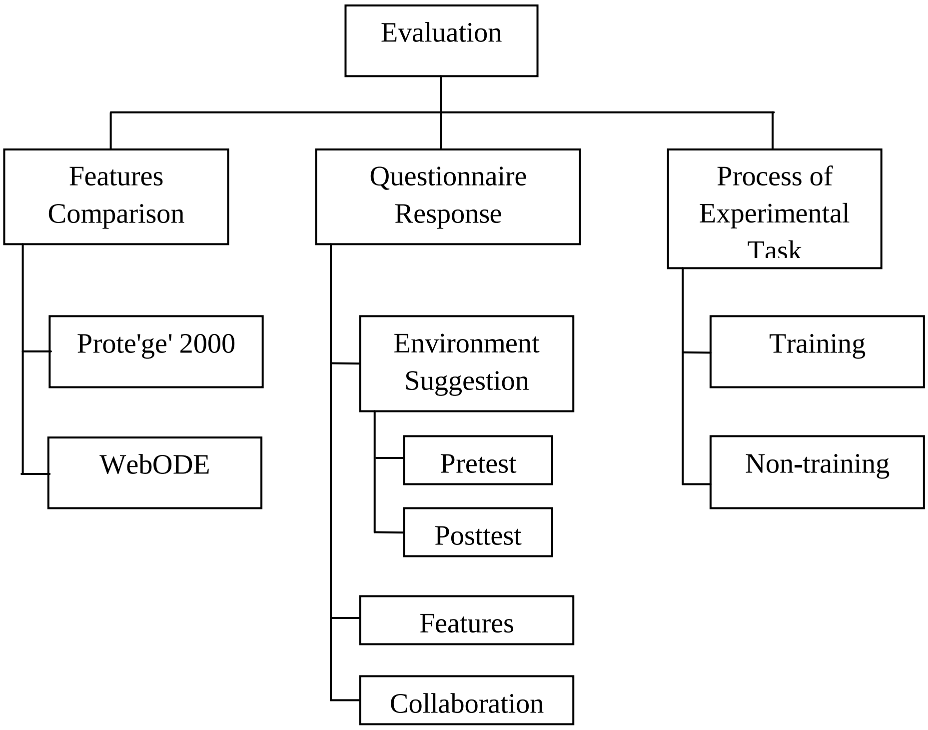
In future, based on this study, the proposed system can be further improved by including schema that provides more complicated knowledge system and strategies for complex explanation in virtual space. Furthermore, implementation in portable device may provide flexibility in access and collaboration at diverse location.

Appendix

Survey of the Proposed system comparison with Protégé 2000 and WebODE.

	Proposed system	Protégé 2000	WebODE
Development tool feature			
- development process	✓	✓	✓
- methodological support	✓	-	✓
- knowledge concept support	✓	✓	✓
- knowledge context support	✓	-	-
- architectural stability	✓	✓	-
- collaborative working	✓	-	✓
- step-wise guidance	✓	-	-
- interface clarity	✓	✓	-
- interface consistency	✓	✓	✓
- Help system	-	-	✓
Merge and integration			
- compliance with standard	-	-	✓
- concept definition	✓	✓	-
- graph structure	✓	✓	-
- instance of concept	✓	✓	-
- language conformity	-	-	✓
- reusable	✓	-	-
- non-local installation	✓	-	✓
Evaluation			
- theory-awareness	-	✓	✓
- concept definition	✓	✓	✓
- consistency checking	✓	✓	✓
- classification	✓	-	-
Annotation			
- extensibility	✓	✓	✓
- change recognition	-	-	-
- libraries	-	✓	✓
- description	✓	✓	✓
Storage and querying			
- import facilities	✓	-	✓
- export facilities	✓	-	-
- backup management	-	-	✓

Experiments Overview



A: Feature Comparison

Subject list

Person no.	First name	Last name	No. of tool experience	Exp. time (months)	Sex	Age
1	ณัชชา	สุขถาวร	2	18	F	25
2	ณัฐกิตติ์	จังพานิช	2	22	M	26
3	วศินี	ปุจฉาการ	2	19	F	26
4	วิชชุวรรณ	สุขไชยศรี	3	25	F	28
5	วิชญาน์นันท์	จิรบวรวณิชย์	2	23	F	27
6	ศุภานุช	มณีเนตร	3	28	F	32
7	ศุภกิตติ์	สุทธิรอด	4	36	M	36

Feature comparison form (translate to English version)

Features Comparison (Proposed system / Protégé 2000 / WebODE)

Name..

Sex : ☐ Male ☐ Female Age.................. (years) KM tool experience.................. (months) Number of used tool .... (tools)

Read the possibilities below and answer to what extent they meet your needs at work, according to the scale on the right.

"+" = positive, e.g. the feature/characteristic is available or properly implemented.
"0" = reasonable, e.g. the feature is available, but it is difficult to use.
"-" = negative, e.g. the feature is not supported or not correctly implemented.
"NA" = not applicable.
"?" = unable to find out.

Questions	(1) Proposed system					(2) Protégé 2000					(3) WebODE				
	+	0	-	NA	?	+	0	-	NA	?	+	0	-	NA	?
q01 Provided features are enough for work															
q02 Meaning of interface is easy to support steps of work															
q03 The clarity by interface is clear															
q04 Interface enable judging facts based on internal parameters															
q05 There is a good overview and particular view															
q06 Does the tool check new data for consistency															
q07 The various concepts in system were well integrated															
q08 The tool allows synchronous editing by different users															
q09 It is possible to import an information from another tool															
q10 It is possible to export an information in various formats															
q11 Provided features are enough for the needs of the corporation															
q12 The changes are made by other users easy to recognize															

----- *Thank you for your cooperation* -----

experiment #1: December 23, 2013 (KMITL)

แบบสอบถามเปรียบเทียบคุณสมบัติของเครื่องมือระหว่าง: ระบบที่นำเสนอ, Protégé 2000, WebODE

ชื่อ-นามสกุล..

เพศ: ☐ ชาย ☐ หญิง **อายุ**.................. (ปี) **ประสบการณ์การใช้เครื่องมือ**.................. (เดือน) **จำนวนเครื่องมือที่เคยใช้งาน** .... (โปรแกรม)

กรุณาตอบคำถามที่ปรากฏอยู่ด้านล่าง โดยเลือกช่องที่อยู่ทางด้านขวา โดยคำถามหนึ่งต่อคำตอบหนึ่งเครื่องมือ

"+" หมายถึง พบ ตามที่ต้องการ
"0" หมายถึง พบ แต่ยังยากต่อการใช้งาน
"-" หมายถึง พบ แต่ไม่สามารถทำงานได้
"NA" หมายถึง ไม่มีปรากฏ
"?" หมายถึง ค้นหาแล้ว แต่ไม่พบ

คำถาม	(1) ระบบที่นำเสนอ					(2) Protégé 2000					(3) WebODE				
	+	0	-	NA	?	+	0	-	NA	?	+	0	-	NA	?
q01 ระบบเตรียมเครื่องมืออย่างเพียงพอในการใช้งาน															
q02 ส่วนติดต่อกับผู้ใช้สนับสนุนอย่างเป็นขั้นเป็นตอนในการปฏิบัติงาน															
q03 ส่วนติดต่อกับผู้ใช้งานมีความชัดเจนและเข้าใจได้															
q04 ส่วนติดต่อกับผู้ใช้มีส่วนช่วยสนับสนุนการตัดสินใจจากปัจจัยภายในระบบ															
q05 ระบบให้มุมมองในภาพรวมและรายละเอียดของข้อมูลได้อย่างชัดเจน															
q06 ระบบมีการตรวจสอบความขัดแย้งของข้อมูลที่เพิ่มเข้าใหม่															
q07 หลักการทำงานของระบบช่วยให้สามารถขยายเพิ่มเติมได้															
q08 ระบบสนับสนุนการทำงานแบบหลายผู้ใช้งานได้															
q09 ระบบสามารถนำเข้าข้อมูลจากเครื่องมืออื่นได้															
q10 ระบบสามารถส่งออกข้อมูลในรูปแบบที่มีความหลากหลาย															
q11 ระบบเอื้ออำนวยและสนับสนุนการทำงานร่วมกันของผู้ใช้ได้															
q12 ระบบมีการจดจำการเปลี่ยนแปลงที่เกิดจากผู้ใช้คนอื่น															

----- *ขอขอบคุณที่ให้ความร่วมมือในการตอบแบบสอบถาม* -----

experiment #1: December 19, 2013 (KMITL)

B: Questionnaire Response

Subject list of advanced user group

Person no.	First name	Last name	Sex	Age	No. of tool experience
1	วศินี	ปุจฉาการ	F	32	3
2	วิชชุวรรณ	สุขไชยศรี	F	27	2
3	สัญชัย	ภักตร์ผ่อง	M	28	2
4	ศุภกิตติ์	สุทธิรอด	M	36	4
5	วิชญาน์นันท์	จิรบวรวณิชย์	F	28	3
6	ศุภานุช	มณีเนตร	F	26	2
7	ณัฐกิตติ์	จังพานิช	M	26	2
8	ณัชชา	สุขถาวร	F	25	2
			M 3, F 5	28.5	2.5

Age average: 25-36 years

Subject list of experienced user group

Person no.	First name	Last name	Sex	Age
1	พิสิษฐ์	เรืองวัฒนกุล	M	21
2	ฟ้าวลัย	ตันศยานนท์	F	22
3	ภัทร์	พูลศิริ	F	21
4	กฤตยชญ์	คงคติธรรม	M	20
5	กฤษกรช์	เอกวรรณัง	M	22
6	ฉัตรพร	ยงทะเล	M	21
7	ปทิตตา	กลิ่นหอม	F	20
8	ปรพิชญ์	เชื้อสุข	M	19
9	ปริญญา	สีมาธรรมรัตน์	M	20
10	ฐิติรัตน์	ศักดิ์พิชัยมงคล	F	21
			M 6, F 4	20.7

Age average: 19-22 years, Tool experience = 1

Subject list of Naïve user group

Person no.	First name	Last name	Sex	Age
1	เชาวนี	จันทร์ทอง	F	37
2	ญานิกา	อินทร์ขำ	F	26
3	ณพวิทย์	เตชเรืองรัศมี	M	31
4	ชญานิศ	ตันธีระพงศ์	F	25
5	ชนกนันท์	เหล่างาม	F	23
6	ยุทธพิชัย	ชาญนิตย์	M	28
7	ยุพเรศ	คงพึ่ง	F	24
8	ณภัทร	ฉัตรชมชื่น	M	28
9	สิทธานต์	รัตนเหลี่ยม	M	24
10	บุณยนุช	มังกรแก้ว	F	22
11	บุษบา	ตันติสุขารมย์	F	24
12	ปฏิพล	สิทธิราพร	M	27
			M 5, F 7	26.583333

Age average: 22-37 years, Tool experience = 0

Result of pretest and posttest of environment-driven suggestion (Q1)

Person no.	Advanced user group		Experienced user group		Naïve user group	
	Pretest	Posttest	Pretest	Posttest	Pretest	Posttest
1	3.125	3.625	2.9	4.1	2.666667	3.666667
2	2.75	3.125	2.7	3.8	2.5	3.583333
3	2.625	3.5	2.2	3.2	2.666667	3.333333
4	2.875	3.375	2.7	3.7	2.25	3.083333
5	2.5	3.5	3.3	3.6	2.333333	3.5
6	2.75	3.125	2.4	3.5	2.583333	3.416667
7	2.375	3.25	2.9	3.6	2.333333	3.083333
8	3.125	3.875	2.5	3.6	2.666667	3.416667
9	2.875	3.25	2.4	3.5	2.166667	3.5
10	3	3	3	3.6	2.083333	2.75
Mean	2.8	3.3625	2.7	3.62	2.425	3.333333
SD	0.251385	0.266471	0.333333	0.229976	0.220304	0.280542

Paired samples statistics (from SPSS)

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Pre-test_advanced	28.0000	8	7.30949	2.58429
	Post-test_advanced	33.6250	8	7.17013	2.53502
Pair 2	Pre-test_experienced	27.0000	10	8.19214	2.59058
	Post-test_experienced	36.2000	10	7.68548	2.43036
Pair 3	Pre-test_naive	24.2500	12	8.99621	2.59698
	Post-test_naive	33.3333	12	8.86601	2.55940

Paired samples correlations (from SPSS)

		N	Correlation	Sig.
Pair 1	pre-test & post-test advanced	8	.979	.000
Pair 2	pre-test & post-test experienced	10	.974	.000
Pair 3	pre-test & post-test naïve	12	.997	.000

Paired samples test (from SPSS)

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	pre-test & post-test advanced	-5.6250	1.50594	.53243	-6.8840	-4.3660	-10.565	7	.000
Pair 2	pre-test & post-test experienced	-9.2000	1.87380	.59255	-10.5404	-7.8596	-15.526	9	.000
Pair 3	pre-test & post-test naïve	-9.0833	.66856	.19300	-9.5081	-8.6586	-47.065	11	.000

Result of basic features questionnaire (Q2)

Person no.	Advanced user group	Experienced user group	Naïve user group
1	4.25	3.7	3.917
2	4.375	4	3.583
3	4	3.9	3.75
4	3.875	3.7	3.667
5	4	3.8	3.5
6	3.25	3.7	3.333
7	3.25	3.5	3.417
8	3	3	3.25
9	3.25	3.8	3.833
10	2.25	2.5	2.583
Mean	3.55	3.56	3.483
SD	0.662	0.462	0.382

Result of collaborative work (Groupware) supportability questionnaire (Q3)

Person no.	Advanced user group	Experienced user group	Naïve user group
1	3.25	3.3	2.5
2	3.875	3.8	3.083
3	3.75	3.5	3.5
4	3.125	2.5	2.833
5	2.25	2.5	2.583
6	2.375	2.3	2.333
7	3.125	3.3	3.583
8	3.875	3.5	3.75
Mean	3.203	3.088	3.021
SD	0.634	0.567	0.541

Sample of user evaluation questionnaire for advanced user group (Q1.A) – English translate

A

Users Evaluation Questionnaire (Q1.A, Q2, Q3)

Experienced on Knowledge and/or Ontology tool: more than 1 tool ....(tools)

Name..

Sex: ☐ Male ☐ Female   Age.................. (years)

Read the possibilities below and answer to what extent they meet your needs at work, according to the scale on the right. In this scale 1 corresponding to "it does not meet the needs" and 5 corresponds to "fully meets the needs".

Q1.A – Pre-test

Environment-driven Suggestion Questions

no.	Questions	Answer Scale
1 e02	There is a good particle view of the information	- ☐☐☐☐☐ + 1 2 3 4 5
2 e04	I found the interface enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
3 e05	I found the interface guide steps of work	- ☐☐☐☐☐ + 1 2 3 4 5
4 e07	The information is easy to apprehend	- ☐☐☐☐☐ + 1 2 3 4 5
5 e09	I think the environment interface encourage my work	- ☐☐☐☐☐ + 1 2 3 4 5
6 e10	I think that I could contribute to this information	- ☐☐☐☐☐ + 1 2 3 4 5
7 e11	I found the various concepts were well integrated	- ☐☐☐☐☐ + 1 2 3 4 5
8 e12	I imagine that most legal experts would understand this information very quickly	- ☐☐☐☐☐ + 1 2 3 4 5
9 e13	The information enable judging adequacy of conclusion	- ☐☐☐☐☐ + 1 2 3 4 5
10 e14	The information enable judging facts based on internal parameters	- ☐☐☐☐☐ + 1 2 3 4 5

Page 1   *experiment #2: December 23, 2013 (KMITL)*

Basic Features Questions (Q2)

no.	Questions	Answer Scale
1 f01	The meaning of the interfaces are clear	- ☐☐☐☐☐ + 1 2 3 4 5
2 f02	I am confident I understand the conceptualization of the tool	- ☐☐☐☐☐ + 1 2 3 4 5
3 f03	I found the information very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
4 f04	Does the tool check new data for consistency	- ☐☐☐☐☐ + 1 2 3 4 5
5 f05	Evaluate the speed of updating after new data is inserted	- ☐☐☐☐☐ + 1 2 3 4 5
6 f06	Are the changes identifiable clear to user	- ☐☐☐☐☐ + 1 2 3 4 5
7 f07	Is it possible to use multiple inheritance	- ☐☐☐☐☐ + 1 2 3 4 5
8 f08	Is it possible to create exhaustive and/or disjoint decomposition	- ☐☐☐☐☐ + 1 2 3 4 5
9 f09	Evaluate the stability of the tool (crashes, etc.)	- ☐☐☐☐☐ + 1 2 3 4 5
10 f10	Are there example available in the tool	- ☐☐☐☐☐ + 1 2 3 4 5

Collaborative Work Supportability Questions (Q3)

no.	Questions	Answer Scale
1 g01	Does the tool allow synchronous editing by different users	- ☐☐☐☐☐ + 1 2 3 4 5
2 g02	Provided features are enough for the needs of the cooperation	- ☐☐☐☐☐ + 1 2 3 4 5
3 g03	I found the sharing of information cumbersome to understand	- ☐☐☐☐☐ + 1 2 3 4 5
4 g04	Are the change made by other user easy to recognize	- ☐☐☐☐☐ + 1 2 3 4 5
5 g05	Are the ways to lock the information	- ☐☐☐☐☐ + 1 2 3 4 5
6 g06	It is possible to browse the information if it locked	- ☐☐☐☐☐ + 1 2 3 4 5
7 g07	It is possible to import an information from another tool	- ☐☐☐☐☐ + 1 2 3 4 5
8 g08	It is possible to export the information in various format	- ☐☐☐☐☐ + 1 2 3 4 5

Page 2

Q1.A – Post-test

Environment-driven Suggestion Questions

no.	Questions	Answer Scale
1 e02	There is a good particle view of the information	- ☐☐☐☐☐ + 1 2 3 4 5
2 e04	I found the interface enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
3 e05	I found the interface guide steps of work	- ☐☐☐☐☐ + 1 2 3 4 5
4 e07	The information is easy to apprehend	- ☐☐☐☐☐ + 1 2 3 4 5
5 e09	I think the environment interface encourage my work	- ☐☐☐☐☐ + 1 2 3 4 5
6 e10	I think that I could contribute to this information	- ☐☐☐☐☐ + 1 2 3 4 5
7 e11	I found the various concepts were well integrated	- ☐☐☐☐☐ + 1 2 3 4 5
8 e12	I imagine that most legal experts would understand this information very quickly	- ☐☐☐☐☐ + 1 2 3 4 5
9 e13	The information enable judging adequacy of conclusion	- ☐☐☐☐☐ + 1 2 3 4 5
10 e14	The information enable judging facts based on internal parameters	- ☐☐☐☐☐ + 1 2 3 4 5

Comment and Suggestion

____

____

____

____

____

____

---- Thank you for your cooperation ---

Page 3

Sample of user evaluation questionnaire for advanced user group (Q1.A) – Thai version

A

<u>แบบสอบถามด้านการทำงานและการสนับสนุนการทำงานร่วมกันของเครื่องมือ (Q1.A, Q2, Q3)</u>

ประสบการณ์ในการใช้เครื่องมือด้านการจัดการความรู้และ/หรือออนโทโลยี: <u>มากกว่า 1 เครื่องมือ</u> ....

ชื่อ-นามสกุล...

เพศ: ☐ ชาย ☐ หญิง อายุ.................. ปี

กรุณาตอบคำถามที่ปรากฏอยู่ด้านล่าง ในช่องระดับคะแนน (1-5) ที่ปรากฏอยู่ทางด้านขวา โดยระดับ 1 หมายถึง ไม่เป็นไปตามที่ต้องการ ไล่ระดับถึง ระดับที่ 5 หมายถึง เป็นไปตามที่ต้องการมากที่สุด

คำถามก่อนการทดสอบ (Q1.A – Pre-test)

คำถามเกี่ยวกับการสนับสนุนการทำงานโดยมีสภาพแวดล้อมเป็นตัวขับเคลื่อน

ข้อ	คำถาม	คำตอบ
1 e02	ระบบสามารถแสดงรายละเอียดของข้อมูลได้อย่างดี	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 e04	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความเพียงพอในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 e05	ฉันพบว่าส่วนติดต่อกับผู้ใช้ช่วยแสดงลำดับขั้นตอนในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 e07	ข้อมูลที่แสดงเข้าใจได้ง่าย	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 e09	ฉันคิดว่าสภาพแวดล้อมในการติดต่อสื่อสารช่วยสนับสนุนการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 e10	ฉันคิดว่าฉันสามารถต่อยอดข้อมูลที่พบได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 e11	ฉันพบว่าหลายหลักการที่มีอยู่สามารถต่อขยายได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 e12	ฉันคิดว่าผู้ที่มีความเชี่ยวชาญสามารถเข้าใจข้อมูลนี้ได้อย่างรวดเร็ว	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
9 e13	ข้อมูลที่ปรากฏอยู่มีความเพียงพอสามารถที่จะสรุปผลได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
10 e14	ข้อมูลที่ปรากฏอยู่สามารถตัดสินข้อเท็จจริงได้จากค่าที่ปรากฏอยู่ภายใน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

experiment #2: December 23, 2013 (KMITL)

หน้า 1

คำถามเกี่ยวกับฟังก์ชั่นการทำงานของระบบ (Q2)

ข้อ	คำถาม	คำตอบ
1 f01	ส่วนติดต่อกับผู้ใช้มีความชัดเจน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 f02	ฉันเชื่อมั่นว่าฉันเข้าใจในหลักการของระบบนี้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 f03	ฉันพบว่าข้อมูลที่ปรากฏในระบบ ง่ายต่อการทำความเข้าใจ	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 f04	ระบบสามารถตรวจสอบความสอดคล้องของข้อมูลใหม่ที่พึ่งนำเข้าระบบได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 f05	ความเร็วของการปรับปรุงข้อมูลหลังจากที่มีการนำเข้าข้อมูลใหม่เป็นไปอย่างรวดเร็ว	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 f06	การเปลี่ยนแปลงข้อมูลสามารถปรากฏได้อย่างชัดเจนต่อผู้ใช้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 f07	ระบบมีความสามารถในการสืบทอดแบบหลายผู้สืบทอด (multiple inheritance)	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 f08	ระบบสามารถแยกระดับลงในรายละเอียดปลีกย่อยได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
9 f09	ระบบมีเสถียรภาพในการทำงาน เช่น ไม่หยุดระหว่างการปฏิบัติงาน เป็นต้น	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
10 f10	ระบบมีตัวอย่างปรากฏเพื่อช่วยในการทำงานของผู้ใช้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

คำถามเกี่ยวกับฟังก์ชั่นการทำงานของระบบ (Q3)

ข้อ	คำถาม	คำตอบ
1 g01	ระบบอนุญาตให้ทำงานแบบไม่ต่อเนื่อง (synchronous editing) ร่วมกันระหว่างผู้ใช้งานได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 g01	ระบบมีการจัดเตรียมวิธีการที่เพียงพอต่อการทำงานร่วมกันกับผู้อื่น	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 g01	ฉันพบว่าระบบสามารถแบ่งปันข้อมูลได้อย่างชัดเจนและเข้าใจได้ง่าย	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 g01	ระบบสามารถบ่งบอกได้ว่าผู้ใช้งานคนอื่น มีการทำการเปลี่ยนแปลงกับข้อมูล	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 g01	ระบบมีการป้องกันข้อมูลในกรณีที่มีผู้ใช้งานคนอื่นทำงานอยู่	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 g01	ระบบสามารถค้นหาได้ว่า ข้อมูลใดมีการป้องกันในกรณีมีผู้ใช้หลายคน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 g01	ระบบสามารถนำเข้าข้อมูลจากเครื่องมืออื่นได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 g01	ระบบสามารถส่งออกข้อมูลได้หลากหลายรูปแบบ	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

หน้า 2

คำถามหลังการทดสอบ (Q1.A – Post-test)

คำถามเกี่ยวกับการสนับสนุนการทำงานโดยมีสภาพแวดล้อมเป็นตัวขับเคลื่อน

ข้อ	คำถาม	คำตอบ
1 e02	ระบบสามารถแสดงรายละเอียดของข้อมูลได้อย่างดี	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 e04	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความเพียงพอในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 e05	ฉันพบว่าส่วนติดต่อกับผู้ใช้ช่วยแสดงลำดับขั้นตอนในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 e07	ข้อมูลที่แสดงเข้าใจได้ง่าย	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 e09	ฉันคิดว่าสภาพแวดล้อมในการติดต่อสื่อสารช่วยสนับสนุนการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 e10	ฉันคิดว่าฉันสามารถต่อยอดข้อมูลที่พบได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 e11	ฉันพบว่าหลายหลักการที่มีอยู่สามารถต่อขยายได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 e12	ฉันคิดว่าผู้ที่มีความเชี่ยวชาญสามารถเข้าใจข้อมูลนี้ได้อย่างรวดเร็ว	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
9 e13	ข้อมูลที่ปรากฏอยู่มีความเพียงพอสามารถที่จะสรุปผลได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
10 e14	ข้อมูลที่ปรากฏอยู่สามารถตัดสินข้อเท็จจริงได้จากค่าที่ปรากฏอยู่ภายใน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

ข้อแสนอแนะเพิ่มเติม

____

____

____

____

____

____

--- ขอบคุณที่ช่วยกรอกแบบสอบถาม --

หน้า 3

Sample of user evaluation questionnaire for experienced user group (Q1.B) – English translate

B

Users Evaluation Questionnaire (Q1.B, Q2, Q3)

Experienced on Knowledge and/or Ontology tool: Only 1 tool

Name..

Sex: ☐ Male ☐ Female Age.................. (years)

Read the possibilities below and answer to what extent they meet your needs at work, according to the scale on the right. In this scale 1 corresponding to "it does not meet the needs" and 5 corresponds to "fully meets the needs".

Q1.B – Pre-test

Environment-driven Suggestion Questions

no.	Questions	Answer Scale
1 e01	There is a good overview of the information	- ☐☐☐☐☐ + 1 2 3 4 5
2 e03	I found the interface easy for work	- ☐☐☐☐☐ + 1 2 3 4 5
3 e04	I found the interface enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
4 e05	I found the interface guide steps of work	- ☐☐☐☐☐ + 1 2 3 4 5
5 e06	The meaning of information is very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
6 e08	The information is enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
7 e09	I think the environment interface encourage my work	- ☐☐☐☐☐ + 1 2 3 4 5
8 e10	I think that I could contribute to this information	- ☐☐☐☐☐ + 1 2 3 4 5
9 e11	I found the various concepts were well integrated	- ☐☐☐☐☐ + 1 2 3 4 5
10 e12	I imagine that most legal experts would understand this information very quickly	- ☐☐☐☐☐ + 1 2 3 4 5

Page 1 *experiment #2: December 23, 2013 (KMITL)*

Basic Features Questions (Q2)

no.	Questions	Answer Scale
1 f01	The meaning of the interfaces are clear	- ☐☐☐☐☐ + 1 2 3 4 5
2 f02	I am confident I understand the conceptualization of the tool	- ☐☐☐☐☐ + 1 2 3 4 5
3 f03	I found the information very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
4 f04	Does the tool check new data for consistency	- ☐☐☐☐☐ + 1 2 3 4 5
5 f05	Evaluate the speed of updating after new data is inserted	- ☐☐☐☐☐ + 1 2 3 4 5
6 f06	Are the changes identifiable clear to user	- ☐☐☐☐☐ + 1 2 3 4 5
7 f07	Is it possible to use multiple inheritance	- ☐☐☐☐☐ + 1 2 3 4 5
8 f08	Is it possible to create exhaustive and/or disjoint decomposition	- ☐☐☐☐☐ + 1 2 3 4 5
9 f09	Evaluate the stability of the tool (crashes, etc.)	- ☐☐☐☐☐ + 1 2 3 4 5
10 f10	Are there example available in the tool	- ☐☐☐☐☐ + 1 2 3 4 5

Collaborative Work Supportability Questions (Q3)

no.	Questions	Answer Scale
1 g01	Does the tool allow synchronous editing by different users	- ☐☐☐☐☐ + 1 2 3 4 5
2 g02	Provided features are enough for the needs of the cooperation	- ☐☐☐☐☐ + 1 2 3 4 5
3 g03	I found the sharing of information cumbersome to understand	- ☐☐☐☐☐ + 1 2 3 4 5
4 g04	Are the change made by other user easy to recognize	- ☐☐☐☐☐ + 1 2 3 4 5
5 g05	Are the ways to lock the information	- ☐☐☐☐☐ + 1 2 3 4 5
6 g06	It is possible to browse the information if it locked	- ☐☐☐☐☐ + 1 2 3 4 5
7 g07	It is possible to import an information from another tool	- ☐☐☐☐☐ + 1 2 3 4 5
8 g08	It is possible to export the information in various format	- ☐☐☐☐☐ + 1 2 3 4 5

Page 2

Q1.B – Post-test

Environment-driven Suggestion Questions

no.	Questions	Answer Scale
1 e01	There is a good overview of the information	- ☐☐☐☐☐ + 1 2 3 4 5
2 e03	I found the interface easy for work	- ☐☐☐☐☐ + 1 2 3 4 5
3 e04	I found the interface enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
4 e05	I found the interface guide steps of work	- ☐☐☐☐☐ + 1 2 3 4 5
5 e06	The meaning of information is very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
6 e08	The information is enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
7 e09	I think the environment interface encourage my work	- ☐☐☐☐☐ + 1 2 3 4 5
8 e10	I think that I could contribute to this information	- ☐☐☐☐☐ + 1 2 3 4 5
9 e11	I found the various concepts were well integrated	- ☐☐☐☐☐ + 1 2 3 4 5
10 e12	I imagine that most legal experts would understand this information very quickly	- ☐☐☐☐☐ + 1 2 3 4 5

Comments and Suggestion

____

---- Thank you for your cooperation ---

Page 3

Sample of user evaluation questionnaire for experienced user group (Q1.B) – Thai version

B

แบบสอบถามด้านการทำงานและการสนับสนุนการทำงานร่วมกันของเครื่องมือ (Q1.B, Q2, Q3)

ประสบการณ์ในการใช้เครื่องมือด้านการจัดการความรู้และ/หรือออนโทโลยี: **1 เครื่องมือ**

ชื่อ-นามสกุล..

เพศ: ☐ ชาย ☐ หญิง อายุ.................. ปี

กรุณาตอบคำถามที่ปรากฏอยู่ด้านล่าง ในช่องระดับคะแนน (1-5) ที่ปรากฏอยู่ทางด้านขวา โดยระดับ 1 หมายถึง ไม่เป็นไปตามที่พบ ไล่ระดับถึง ระดับที่ 5 หมายถึง เป็นไปตามที่พบมากที่สุด

คำถามก่อนการทดสอบ (Q1.B – Pre-test)

คำถามเกี่ยวกับการสนับสนุนการทำงานโดยมีสภาพแวดล้อมเป็นตัวขับเคลื่อน

ข้อ	คำถาม	คำตอบ
1 e01	ระบบสามารถแสดงภาพรวมของข้อมูลได้อย่างดี	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 e03	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความง่ายในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 e04	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความเพียงพอในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 e05	ฉันพบว่าส่วนติดต่อกับผู้ใช้ช่วยแสดงขั้นลำดับของการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 e06	ความหมายของข้อมูลที่ปรากฏในระบบสามารถเข้าใจได้ง่าย	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 e08	ข้อมูลมีความเพียงพอในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 e09	ฉันคิดว่าสภาพแวดล้อมในการติดต่อสื่อสารช่วยสนับสนุนการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 e10	ฉันคิดว่าฉันสามารถต่อยอดข้อมูลที่มีอยู่ได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
9 e11	ฉันพบว่าหลายหลักการที่มีอยู่สามารถต่อขยายได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
10 e12	ฉันคิดว่าผู้ที่มีความเชี่ยวชาญสามารถเข้าใจข้อมูลนี้ได้อย่างรวดเร็ว	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

experiment #2: December 23, 2013 (KMITL)

หน้า 1

คำถามเกี่ยวกับฟังก์ชั่นการทำงานของระบบ (Q2)

ข้อ	คำถาม	คำตอบ
1 f01	ส่วนติดต่อกับผู้ใช้มีความชัดเจน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 f02	ฉันเชื่อมั่นว่าฉันเข้าใจในหลักการของระบบนี้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 f03	ฉันพบว่าข้อมูลที่ปรากฏในระบบ ง่ายต่อการทำความเข้าใจ	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 f04	ระบบสามารถตรวจสอบความสอดคล้องของข้อมูลใหม่ที่พึ่งนำเข้าระบบได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 f05	ความเร็วของการปรับปรุงข้อมูลหลังจากที่มีการนำเข้าข้อมูลใหม่เป็นไปอย่างรวดเร็ว	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 f06	การเปลี่ยนแปลงข้อมูลสามารถปรากฏได้อย่างชัดเจนต่อผู้ใช้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 f07	ระบบมีความสามารถในการสืบทอดแบบหลายผู้สืบทอด (multiple inheritance)	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 f08	ระบบสามารถแยกระดับลงในรายละเอียดปลีกย่อยได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
9 f09	ระบบมีเสถียรภาพในการทำงาน เช่น ไม่หยุดระหว่างการปฏิบัติงาน เป็นต้น	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
10 f10	ระบบมีตัวอย่างปรากฏเพื่อช่วยในการทำงานของผู้ใช้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

คำถามเกี่ยวกับฟังก์ชั่นการทำงานของระบบ (Q3)

ข้อ	คำถาม	คำตอบ
1 g01	ระบบอนุญาตให้ทำงานแบบไม่ต่อเนื่อง (synchronous editing) ร่วมกันระหว่างผู้ใช้งานได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 g01	ระบบมีการจัดเตรียมวิธีการที่เพียงพอต่อการทำงานร่วมกันกับผู้อื่น	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 g01	ฉันพบว่าระบบสามารถแบ่งปันข้อมูลได้อย่างชัดเจนและเข้าใจได้ง่าย	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 g01	ระบบสามารถบ่งบอกได้ว่าผู้ใช้งานคนอื่น มีการทำการเปลี่ยนแปลงกับข้อมูล	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 g01	ระบบมีการป้องกันข้อมูลในกรณีที่มีผู้ใช้งานคนอื่นทำงานอยู่	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 g01	ระบบสามารถค้นหาได้ว่า ข้อมูลใดมีการป้องกันในกรณีมีผู้ใช้หลายคน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 g01	ระบบสามารถนำเข้าข้อมูลจากเครื่องมืออื่นได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 g01	ระบบสามารถส่งออกข้อมูลได้หลากหลายรูปแบบ	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

หน้า 2

คำถามหลังการทดสอบ (Q1.B – Post-test)

คำถามเกี่ยวกับการสนับสนุนการทำงานโดยมีสภาพแวดล้อมเป็นตัวขับเคลื่อน

ข้อ	คำถาม	คำตอบ
1 e01	ระบบสามารถแสดงภาพรวมของข้อมูลได้อย่างดี	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
2 e03	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความง่ายในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
3 e04	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความเพียงพอในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
4 e05	ฉันพบว่าส่วนติดต่อกับผู้ใช้ช่วยแสดงขั้นลำดับของการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
5 e06	ความหมายของข้อมูลที่ปรากฏในระบบสามารถเข้าใจได้ง่าย	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
6 e08	ข้อมูลมีความเพียงพอในการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
7 e09	ฉันคิดว่าสภาพแวดล้อมในการติดต่อสื่อสารช่วยสนับสนุนการทำงาน	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
8 e10	ฉันคิดว่าฉันสามารถต่อยอดข้อมูลที่มีอยู่ได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
9 e11	ฉันพบว่าหลายหลักการที่มีอยู่สามารถต่อขยายได้	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5
10 e12	ฉันคิดว่าผู้ที่มีความเชี่ยวชาญสามารถเข้าใจข้อมูลนี้ได้อย่างรวดเร็ว	- ☐ ☐ ☐ ☐ ☐ + 1 2 3 4 5

ข้อเสนอแนะเพิ่มเติม

--- ขอบคุณที่ช่วยกรอกแบบสอบถาม --

หน้า 3

Sample of user evaluation questionnaire for naïve user group (Q1.C) – English translate

C

Users Evaluation Questionnaire (Q1.C, Q2, Q3)

Experienced on Knowledge and/or Ontology tool: Never Used

Name...

Sex: ☐ Male ☐ Female Age.................. (years)

Read the possibilities below and answer to what extent they meet your needs at work, according to the scale on the right. In this scale 1 corresponding to "it does not meet the needs" and 5 corresponds to "fully meets the needs".

Q1.C – Pre-test

Environment-driven Suggestion Questions

no.	Questions	Answer Scale
1 e01	There is a good overview of the information	- ☐☐☐☐☐ + 1 2 3 4 5
2 e02	There is a good particle view of the information	- ☐☐☐☐☐ + 1 2 3 4 5
3 e03	I found the interface easy for work	- ☐☐☐☐☐ + 1 2 3 4 5
4 e04	I found the interface enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
5 e05	I found the interface guide the steps of work	- ☐☐☐☐☐ + 1 2 3 4 5
6 e06	The meaning of information is very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
7 e07	The information is easy to apprehend	- ☐☐☐☐☐ + 1 2 3 4 5
8 e08	The information is enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
9 e09	I think the environment interface encourage my work	- ☐☐☐☐☐ + 1 2 3 4 5
10 e10	I think that I could contribute to this information	- ☐☐☐☐☐ + 1 2 3 4 5

Page 1 *experiment #2: December 23, 2013 (KMITL)*

Basic Features Questions (Q2)

no.	Questions	Answer Scale
1 f01	The meaning of the interfaces are clear	- ☐☐☐☐☐ + 1 2 3 4 5
2 f02	I am confident I understand the conceptualization of the tool	- ☐☐☐☐☐ + 1 2 3 4 5
3 f03	I found the information very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
4 f04	Does the tool check new data for consistency	- ☐☐☐☐☐ + 1 2 3 4 5
5 f05	Evaluate the speed of updating after new data is inserted	- ☐☐☐☐☐ + 1 2 3 4 5
6 f06	Are the changes identifiable clear to user	- ☐☐☐☐☐ + 1 2 3 4 5
7 f07	Is it possible to use multiple inheritance	- ☐☐☐☐☐ + 1 2 3 4 5
8 f08	Is it possible to create exhaustive and/or disjoint decomposition	- ☐☐☐☐☐ + 1 2 3 4 5
9 f09	Evaluate the stability of the tool (crashes, etc.)	- ☐☐☐☐☐ + 1 2 3 4 5
10 f10	Are there example available in the tool	- ☐☐☐☐☐ + 1 2 3 4 5

Collaborative Work Supportability Questions (Q3)

no.	Questions	Answer Scale
1 g01	Does the tool allow synchronous editing by different users	- ☐☐☐☐☐ + 1 2 3 4 5
2 g02	Provided features are enough for the needs of the cooperation	- ☐☐☐☐☐ + 1 2 3 4 5
3 g03	I found the sharing of information cumbersome to understand	- ☐☐☐☐☐ + 1 2 3 4 5
4 g04	Are the change made by other user easy to recognize	- ☐☐☐☐☐ + 1 2 3 4 5
5 g05	Are the ways to lock the information	- ☐☐☐☐☐ + 1 2 3 4 5
6 g06	It is possible to browse the information if it locked	- ☐☐☐☐☐ + 1 2 3 4 5
7 g07	It is possible to import an information from another tool	- ☐☐☐☐☐ + 1 2 3 4 5
8 g08	It is possible to export the information in various format	- ☐☐☐☐☐ + 1 2 3 4 5

Page 2

Q1.C – Post-test

Environment-driven Suggestion Questions

no.	Questions	Answer Scale
1 e01	There is a good overview of the information	- ☐☐☐☐☐ + 1 2 3 4 5
2 e02	There is a good particle view of the information	- ☐☐☐☐☐ + 1 2 3 4 5
3 e03	I found the interface easy for work	- ☐☐☐☐☐ + 1 2 3 4 5
4 e04	I found the interface enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
5 e05	I found the interface guide the steps of work	- ☐☐☐☐☐ + 1 2 3 4 5
6 e06	The meaning of information is very easy to understand	- ☐☐☐☐☐ + 1 2 3 4 5
7 e07	The information is easy to apprehend	- ☐☐☐☐☐ + 1 2 3 4 5
8 e08	The information is enough for work	- ☐☐☐☐☐ + 1 2 3 4 5
9 e09	I think the environment interface encourage my work	- ☐☐☐☐☐ + 1 2 3 4 5
10 e10	I think that I could contribute to this information	- ☐☐☐☐☐ + 1 2 3 4 5

Comments and Suggestion

---- Thank you for your cooperation ---

Page 3

Sample of user evaluation questionnaire for naïve user group (Q1.C) – Thai version

C

<u>แบบสอบถามด้านการทำงานและการสนับสนุนการทำงานร่วมกันของเครื่องมือ (Q1.C, Q2, Q3)</u>

ประสบการณ์ในการใช้เครื่องมือด้านการจัดการความรู้และ/หรือออนโทโลยี: **<u>ไม่มีประสบการณ์</u>**

ชื่อ-นามสกุล...

เพศ: □ ชาย □ หญิง อายุ................. ปี

กรุณาตอบคำถามที่ปรากฏอยู่ด้านล่าง ในช่องระดับคะแนน (1-5) ที่ปรากฏอยู่ทางด้านขวา โดยระดับ 1 หมายถึง ไม่เป็นไปตามที่พบ ไล่ระดับถึง ระดับที่ 5 หมายถึง เป็นไปตามที่พบมากที่สุด

คำถามก่อนการทดสอบ (Q1.C – Pre-test)

คำถามเกี่ยวกับการสนับสนุนการทำงานโดยมีสภาพแวดล้อมเป็นตัวขับเคลื่อน

ข้อ	คำถาม	คำตอบ
1 e01	ระบบสามารถแสดงภาพรวมของข้อมูลได้อย่างดี	- □ □ □ □ □ + (1 2 3 4 5)
2 e02	ระบบสามารถแสดงรายละเอียดของข้อมูลได้อย่างดี	- □ □ □ □ □ + (1 2 3 4 5)
3 e03	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความง่ายในการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
4 e04	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความเพียงพอในการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
5 e05	ฉันพบว่าส่วนติดต่อกับผู้ใช้ช่วยแสดงขั้นลำดับของการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
6 e06	ความหมายของข้อมูลที่ปรากฏสามารถเข้าใจได้ง่าย	- □ □ □ □ □ + (1 2 3 4 5)
7 e07	ข้อมูลที่มีอยู่มีความง่ายต่อการทำความเข้าใจ	- □ □ □ □ □ + (1 2 3 4 5)
8 e08	ข้อมูลที่มีอยู่มีความเพียงพอในการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
9 e09	ฉันคิดว่าสภาพแวดล้อมในการติดต่อสื่อสารช่วยสนับสนุนการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
10 e10	ฉันคิดว่าฉันสามารถต่อยอดข้อมูลที่มีอยู่ได้	- □ □ □ □ □ + (1 2 3 4 5)

หน้า 1 *experiment #2: December 23, 2013 (KMITL)*

คำถามเกี่ยวกับฟังก์ชั่นการทำงานของระบบ (Q2)

ข้อ	คำถาม	คำตอบ
1 f01	ส่วนติดต่อกับผู้ใช้มีความชัดเจน	- □ □ □ □ □ + (1 2 3 4 5)
2 f02	ฉันเชื่อมั่นว่าฉันเข้าใจในหลักการของระบบนี้	- □ □ □ □ □ + (1 2 3 4 5)
3 f03	ฉันพบว่าข้อมูลที่ปรากฏในระบบ ง่ายต่อการทำความเข้าใจ	- □ □ □ □ □ + (1 2 3 4 5)
4 f04	ระบบสามารถตรวจสอบความสอดคล้องของข้อมูลใหม่ที่พึ่งนำเข้าระบบได้	- □ □ □ □ □ + (1 2 3 4 5)
5 f05	ความเร็วของการปรับปรุงข้อมูลหลังจากที่มีการนำเข้าข้อมูลใหม่เป็นไปอย่างรวดเร็ว	- □ □ □ □ □ + (1 2 3 4 5)
6 f06	การเปลี่ยนแปลงข้อมูลสามารถปรากฏได้อย่างชัดเจนต่อผู้ใช้	- □ □ □ □ □ + (1 2 3 4 5)
7 f07	ระบบมีความสามารถในการสืบทอดแบบหลายผู้สืบทอด (multiple inheritance)	- □ □ □ □ □ + (1 2 3 4 5)
8 f08	ระบบสามารถแยกระดับลงในรายละเอียดปลีกย่อยได้	- □ □ □ □ □ + (1 2 3 4 5)
9 f09	ระบบมีเสถียรภาพในการทำงาน เช่น ไม่หยุดระหว่างการปฏิบัติงาน เป็นต้น	- □ □ □ □ □ + (1 2 3 4 5)
10 f10	ระบบมีตัวอย่างปรากฏเพื่อช่วยในการทำงานของผู้ใช้	- □ □ □ □ □ + (1 2 3 4 5)

คำถามเกี่ยวกับฟังก์ชั่นการทำงานของระบบ (Q3)

ข้อ	คำถาม	คำตอบ
1 g01	ระบบอนุญาตให้ทำงานแบบไม่ต่อเนื่อง (synchronous editing) ร่วมกันระหว่างผู้ใช้งานได้	- □ □ □ □ □ + (1 2 3 4 5)
2 g01	ระบบมีการจัดเตรียมวิธีการที่เพียงพอต่อการทำงานร่วมกันกับผู้อื่น	- □ □ □ □ □ + (1 2 3 4 5)
3 g01	ฉันพบว่าระบบสามารถแบ่งปันข้อมูลได้อย่างชัดเจนและเข้าใจได้ง่าย	- □ □ □ □ □ + (1 2 3 4 5)
4 g01	ระบบสามารถบ่งบอกได้ว่าผู้ใช้งานคนอื่น มีการทำการเปลี่ยนแปลงกับข้อมูล	- □ □ □ □ □ + (1 2 3 4 5)
5 g01	ระบบมีการป้องกันข้อมูลในกรณีที่มีผู้ใช้งานคนอื่นทำงานอยู่	- □ □ □ □ □ + (1 2 3 4 5)
6 g01	ระบบสามารถค้นหาได้ว่า ข้อมูลใดมีการป้องกันในกรณีมีผู้ใช้หลายคน	- □ □ □ □ □ + (1 2 3 4 5)
7 g01	ระบบสามารถนำเข้าข้อมูลจากเครื่องมืออื่นได้	- □ □ □ □ □ + (1 2 3 4 5)
8 g01	ระบบสามารถส่งออกข้อมูลได้หลากหลายรูปแบบ	- □ □ □ □ □ + (1 2 3 4 5)

หน้า 2

คำถามหลังการทดสอบ (Q1.C – Post-test)

คำถามเกี่ยวกับการสนับสนุนการทำงานโดยมีสภาพแวดล้อมเป็นตัวขับเคลื่อน

ข้อ	คำถาม	คำตอบ
1 e01	ระบบสามารถแสดงภาพรวมของข้อมูลได้อย่างดี	- □ □ □ □ □ + (1 2 3 4 5)
2 e02	ระบบสามารถแสดงรายละเอียดของข้อมูลได้อย่างดี	- □ □ □ □ □ + (1 2 3 4 5)
3 e03	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความง่ายในการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
4 e04	ฉันพบว่าส่วนติดต่อกับผู้ใช้มีความเพียงพอในการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
5 e05	ฉันพบว่าส่วนติดต่อกับผู้ใช้ช่วยแสดงขั้นลำดับของการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
6 e06	ความหมายของข้อมูลที่ปรากฏสามารถเข้าใจได้ง่าย	- □ □ □ □ □ + (1 2 3 4 5)
7 e07	ข้อมูลที่มีอยู่มีความง่ายต่อการทำความเข้าใจ	- □ □ □ □ □ + (1 2 3 4 5)
8 e08	ข้อมูลที่มีอยู่มีความเพียงพอในการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
9 e09	ฉันคิดว่าสภาพแวดล้อมในการติดต่อสื่อสารช่วยสนับสนุนการทำงาน	- □ □ □ □ □ + (1 2 3 4 5)
10 e10	ฉันคิดว่าฉันสามารถต่อยอดข้อมูลที่มีอยู่ได้	- □ □ □ □ □ + (1 2 3 4 5)

ข้อเสนอแนะเพิ่มเติม

____

____

____

____

____

____

--- ขอบคุณที่ช่วยกรอกแบบสอบถาม --

หน้า 3

C: Process of Experimental Tasks

Training user group

No.	First name	Last name	Sex	Age
1	กนกวรรณ	มุตตามระ	F	20
2	กลทีป	พยุหวรรธนะ	M	20
3	กษิดิศ	ปิยธรรมวงศ์	M	19
4	กันต์ฤทัย	ประเสริฐพันธุ์	F	20
5	กันต์	ภู่โชติแสงสวัสดิ์	M	20

Male 3: Female 2, age average: 19.8 years

Non-training user group

No.	First name	Last name	Sex	Age
1	กิตติกร	ประเสริฐศักดิ์	M	21
2	กิติศักดิ์	แสนโท	M	19
3	กุลศิตา	ดอนฉิมพลี	F	20
4	เขมทัต	เล่งไพบูลย์	M	20
5	คณิต	โพธิ์อาศัย	M	20

Male 4: Female 1, age average: 20 years

Process of experimental task of training user group with spending time (unit: seconds).

Person No.	Task 1	Task 2	Task 3
1	65	123	164
2	78	115	158
3	83	134	189
4	72	136	177
5	85	148	195
Average	76.6	131.2	176.6
SD	8.203658	12.67675	15.78924

Process of experimental task of non-training user group with spending time (unit: seconds).

Person No.	Task 1		Task 2		Task 3	
	Time #1	Time #2	Time #1	Time #2	Time #1	Time #2
1	102	85	167	158	210	185
2	98	64	210	172	196	196
3	108	82	188	138	249	188
4	115	68	148	143	188	164
5	128	74	161	128	173	178
Average	110.2	74.6	174.8	147.8	203.2	182.2
SD	11.84061	8.933085	24.40697	17.32628	28.89118	12.0499

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	PE_G1	22.4000	10	2.01108	.63596
	PO_G1	26.9000	10	2.13177	.67412
Pair 2	PE_G2	27.0000	10	3.33333	1.05409
	PO_G2	36.2000	10	2.29976	.72725
Pair 3	PE_G3	29.1000	10	2.64365	.83600
	PO_G3	40.0000	10	3.36650	1.06458
Pair 4	TASK1_TI	76.6000	5	8.20366	3.66879
	TASK1_#1	110.2000	5	11.84061	5.29528
Pair 5	TASK1_TI	76.6000	5	8.20366	3.66879
	TASK1_#2	74.6000	5	8.93308	3.99500
Pair 6	TASK2_TI	131.2000	5	12.67675	5.66922
	TASK2_#1	174.8000	5	24.40697	10.91513
Pair 7	TASK2_TI	131.2000	5	12.67675	5.66922
	TASK2_#2	147.8000	5	17.32628	7.74855
Pair 8	TASK3_TI	176.6000	5	15.78924	7.06116
	TASK3_#1	203.2000	5	28.89118	12.92053
Pair 9	TASK3_TI	176.6000	5	15.78924	7.06116
	TASK3_#2	182.2000	5	12.04990	5.38888

*** pair 4-9 only**

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	PE_G1 & PO_G1	10	.295	.407
Pair 2	PE_G2 & PO_G2	10	.522	.122
Pair 3	PE_G3 & PO_G3	10	.612	.060
Pair 4	TASK1_TI & TASK1_#1	5	.493	.399
Pair 5	TASK1_TI & TASK1_#2	5	-.214	.729
Pair 6	TASK2_TI & TASK2_#1	5	-.670	.215
Pair 7	TASK2_TI & TASK2_#2	5	-.978	.004
Pair 8	TASK3_TI & TASK3_#1	5	.030	.962
Pair 9	TASK3_TI & TASK3_#2	5	-.400	.504

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	PE_G1 - PO_G1	-4.5000	2.46080	.77817	-6.2604	-2.7396	-5.783	9	.000
Pair 2	PE_G2 - PO_G2	-9.2000	2.89828	.91652	-11.2733	-7.1267	-10.038	9	.000
Pair 3	PE_G3 - PO_G3	-10.9000	2.72641	.86217	-12.8504	-8.9496	-12.643	9	.000
Pair 4	TASK1_TI - TASK1_#1	-33.6000	10.57355	4.72864	-46.7288	-20.4712	-7.106	4	.002
Pair 5	TASK1_TI - TASK1_#2	2.0000	13.36039	5.97495	-14.5891	18.5891	.335	4	.755
Pair 6	TASK2_TI - TASK2_#1	-43.6000	34.22426	15.30555	-86.0950	-1.1050	-2.849	4	.046
Pair 7	TASK2_TI - TASK2_#2	-16.6000	29.83790	13.34391	-53.6486	20.4486	-1.244	4	.281
Pair 8	TASK3_TI - TASK3_#1	-26.6000	32.50846	14.53823	-66.9646	13.7646	-1.830	4	.141
Pair 9	TASK3_TI - TASK3_#2	-5.6000	23.38376	10.45753	-34.6348	23.4348	-.535	4	.621
