東海大学紀要情報通信学部
vol.5,No1,2012,pp.13-19

Paper

Knowledge Memory System Design on Virtual Space

by

Boonprasert SURAKRATANASAKUL[*1] and Kazuhiko HAMAMOTO[*2]

(received on April 6, 2012 & accepted on July 3, 2012)

Abstract

On knowledge management strategies, knowledge engineer needs to explore a lot of past knowledge according to a current situation and then pile the new ideas upon. The difficultly is to find an appropriate way for managing and maintenance huge information on a schematic way.

Knowledge memory system on virtual space is a scenic representation of huge information that is organized spatially and temporally. It enables knowledge engineer to visually grasp and explore the information space for accommodate new information at an appropriate place based on manner understandable in knowledge management concept. We apply two schemas for knowledge space with CommonKADS concept, are called "Knowledge Landscape" and "Knowledge Atlas". Knowledge landscape schema focus on capture and investigate on related group of knowledge structure. Knowledge atlas schema focus on understand of an introduction to enterprise knowledge. Both of two schemas are correlated with "Knowledge Card" that consists of meta-knowledge and hyperlink information within XML file.

Keywords: knowledge memory system, knowledge modeling, knowledge management, knowledge landscape, knowledge atlas.

1. Introduction

Knowledge Management is an important way for managing knowledge of an organization that embedded in people, processes and information generators. It views knowledge as a valuable asset for value-added organization. Many organizations define their own knowledge management strategies for explicating, developing and distributing their knowledge in order to stay competitive and be innovative [1].

One of the purposes of knowledge management is that people could share their explicit and/or tacit knowledge with others in an organization. In order to achieve this goal, knowledge engineer needs to discover the past knowledge information according to a present situation and then pile new ideas upon them, as known as synergy knowledge mechanism [2]. Unfortunately, it is not easy manageable if they lack of a good overview for give a big picture in an understandable manner.

From above reason, this paper purposes an approach for design the knowledge memory system on the virtual space. This system is a scenic representation of huge information that is organized spatially and temporally on a virtual space. It enables users to visually grasp the global nature of knowledge, explore the information space, and accommodate new information at an appropriate place based on users understandable [3, 4].

To develop the system, we create two schemas for modeling the knowledge virtual space. They are applied from the CommonKADS methodology concept in two levels: Knowledge level and Organization knowledge level. For Knowledge level, we develop a schema name "Knowledge Landscape". On the other hand, we develop schema name "Knowledge Atlas" for Organization knowledge level. Each of level has an individual perspective space for managing and maintains knowledge within their concept, and both are correlated with the "Knowledge Card" that represent knowledge item in domain.

To address issues, the rest of the paper is organized as follows: section 2 briefly reviews about the relevant concepts: the CommonKADS background concept with the Knowledge level in subsection 2.1 and the Organization knowledge level in subsection 2.2. Subsection 2.3 describes an architectural

*1 Graduate School of Science and Tech., Tokai Univ.
*2 Dept. Infor. & Media Tech., School of Infor. and Telecommunication Eng., Tokai Univ.

model view and UML extension mechanism that are background mechanism for develop the knowledge schema. Section 3 describes two knowledge schemas and their definitions: the Knowledge Landscape in subsection 3.1 and the Knowledge Atlas in subsection 3.2. For section 4, we describe framework of the knowledge memory system on virtual space, the knowledge representation, and the topological of knowledge. Section 5 describes the knowledge scenario development life cycle and an example. Finally, section 6 is the conclusions and future works.

2. Relevant Concepts

2.1 *CommonKADS's Knowledge Level*

CommonKADS methodology originates from the need to build the industry-quality knowledge systems on a large scale, in a structured, controllable, and repeatable way [5]. It helps to clarify the structure of a knowledge-intensive task and provides a specification of data and knowledge structures. It has an essence structure that similar to analysis models in software engineering [6]. Meanwhile we can use the standard of UML to develop the knowledge schema model. Generally, CommonKADS methodology concept consists of three levels but in this research, we use only two levels for develop the purposed system: Knowledge level and Organization knowledge level.

In CommonKADS's knowledge level, it consists of three knowledge category that each capturing a related group of knowledge structure: Domain knowledge, Inference knowledge, and Task knowledge.

- Domain knowledge 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 the application domain. Ordinarily, domain knowledge description consists of two types: Domain schema and Knowledge base [7].

- Inference knowledge describes the basic inference steps that want to make using domain knowledge. It describes how these static structures can be used to carry out a reasoning process. In software engineering term, Inferences represent the lowest level of functional decomposition [8]. The main ingredients of the inference knowledge are: Inference, Knowledge role, and Transfer function.

- 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 similar to the higher levels of functional decomposition in software engineering, but also includes control over the functions involved [9]. Task can be decomposed into subtasks or into basic inferences. At the lowest level of task decomposition, the tasks are linked to the inferences and transfer functions.

2.2 *CommonKADS's Organization Knowledge Level*

CommonKADS methodology is one of prominent technique for managing organization knowledge. It provides an extensive method for describing business processes in which knowledge-intensive tasks are carried out. It deploys as the technological means for capturing and managing both explicit and tacit knowledge as part of knowledge management initiative. The capabilities of CommonKADS were no longer limited to the emulation of expert reasoning; they could also be applied to managing the enterprise knowledge such as, business rules, procedures, and guidelines.

In organization knowledge level, CommonKADS provides the series of worksheets to support knowledge engineer for develop and understand an organizational aspect. The study is conducted based on problems and opportunities. It can focus on such areas as: structure, process, people, culture and human power bases, resources, process breakdowns, and knowledge assets [10].

2.3 *Architectural Model View and UML Extension Mechanism*

In CommonKADS procedure, knowledge engineer explains the knowledge structure by the modeling. Domain knowledge model is designed with the notation similar to UML class. Task and Inference knowledge have own graphical notations, as nondirective UML equivalent [11]. These appear an ambiguous to understand on different the notation meaning. On the other hand, in organization knowledge level provides the suite of worksheets for developing an organization aspect. Nevertheless, they lack of a good understanding in a comprehensive picture of how an organization uses its knowledge [12].

For solving above problems, we apply the concept of architectural model view and UML extension mechanism for all artifacts to be one standard for creating the knowledge schemas.

- Architectural model view deals with abstraction, with decomposition and composition, with style and esthetics. It use a model composed of multiple views to describe the architecture [13]. In research, we use only two views that related to the knowledge paradigm for creating the knowledge schemas: Logical view and Functional view.

Logical view is a primarily supports what the system should provide in terms of services. In this view, the system is decomposed into a set of key abstractions, taken mostly from the domain in the form of objects or object classes. For the functional view, it supports how the system's elements work together seamlessly by usage scenario. We use this view to describe the corresponding script like the inference structure and task knowledge scenario.

- UML extension mechanism is provided by the UML standard in order to allow users to customize and extend the language to suite their particular needs. It consists of three types: Stereotype, Tagged value, and Constraint [14].

Stereotype is an extension of the vocabulary which allows designer to create new building blocks from existing ones but specific to the domain problem. General form of stereotype is either <<stereotype-name>> or special (user-defined) stereotype icons. Tagged value is an extension of the properties which allows designer to create new information in that element specification. General form of tagged value is: {tag=value}. Lastly, Constraint is an extension of the semantics. It represents rules that are applied to model. Designer can employ both predefined and user-defined constraints. Constraints may also be defined using the Object Constraint Language (OCL) [15].

3. Knowledge Schemas

Knowledge schema is a pattern that creates for explanation the concept of knowledge model on virtual space. It has two schemas in research: Knowledge Landscape schema and Knowledge Atlas schema.

3.1 *Knowledge Landscape Schema*

Knowledge landscape schema focus on capture and investigate on the knowledge category. It helps to clarify the structure of a knowledge-intensive task and provide a specification of data and the knowledge structure. Knowledge landscape schema consists of two perspective views: logical view and functional view. Logical view provides abstract for representation of domain knowledge and knowledge base. Functional view realizes the scenarios from the knowledge-intensive task that correspond with the inference and task knowledge by Inside-Out and Outside-In realization [16]. Fig.1 shows the knowledge landscape schema for knowledge level. The description of schema elements are itemized in Table 1.

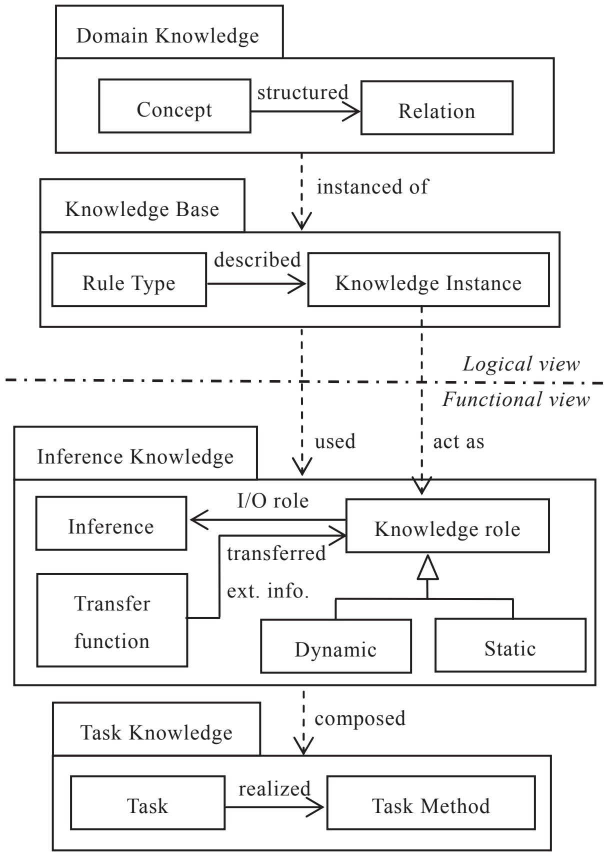

Fig.1 The Knowledge Landscape Schema.

Table 1 Element Description of Knowledge Landscape Schema.

Element	Description
Domain Knowledge	Package contains the domain specific knowledge and information types.
Knowledge Base	Package contains the instances of types specified in domain schema.
Inference Knowledge	Package contains the specification of invocation of inference method.
Task Knowledge	Package contains the suite of reasoning function.
Concept	Metaclass represents the category of things with abstract concept.
Relation	Metaclass used for more complicated types of modeling and defined through a specification of arguments.
Rule type	Metaclass used for categorization and specification of knowledge.
Knowledge Instance	Metaclass of instances of domain knowledge concept.

Element	Description
Inference	Metaclass describes the lowest level of functional decomposition on carrying out primitive reasoning steps.
Knowledge role	Metaclass defines the functional roles of knowledge in reasoning process.
Dynamic knowledge role	Type of knowledge role in run-time inputs and outputs of inferences.
Static knowledge role	Type of knowledge role that specifies the collection of domain knowledge that is used to make the inference.
Transfer function	Functions transfer information between reasoning agent and external entities.
Task	Metacalss defines the reasoning function and invokes the corresponding task method.
Task Method	Formalize method control structure.

3.2 *Knowledge Atlas Schema*

Knowledge atlas is a schema that captures the essential features of real system. It starts with an understanding of the broader an organizational aspect. It argues that models are important for understanding working mechanisms within context and environment of knowledge-based system, such as: organization aspect, agents, and tasks. In addition, it contributes the understanding of the sources of knowledge, inputs and outputs, flow of knowledge and identification of other variables such as the impact that management action has on the knowledge of organization.

Although different organization's systems have different goals and internal structures, they use similar concepts to describe their structure and operations [17]. Fig.2 shows the schema of knowledge atlas for organization knowledge level. The description of schema elements are itemized in Table 2.

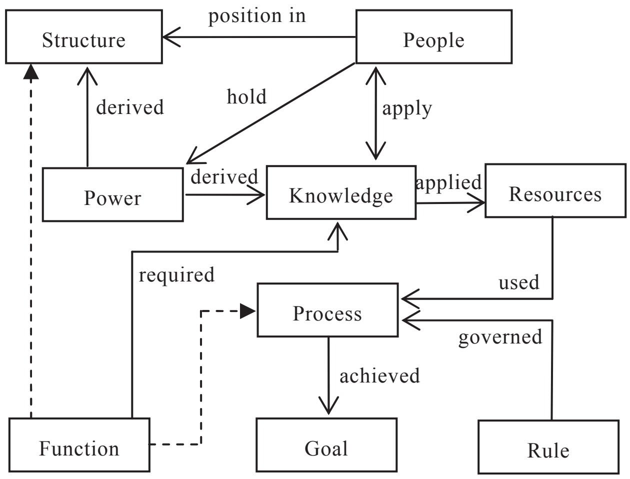

Fig.2 The Knowledge Atlas Schema.

Table 2 Element Decription of Knowledge Atlas Schema.

Element	Description
Structure	Metaclass represents an organization is built from structural units.
Function	Each structural unit carries out one or more business functions.
Process	Process describes how the work is done within business.
People	People play roles in the organization and fill positions in Structure.
Power	People derive power from their role and knowledge.
Resources	The objects are used or produced in the business.
Knowledge	Knowledge items characterize the knowledge in the organization at a fairy general level of description, mainly for managerial purposes.
Goal	The purpose of the business or the outcome is trying to achieve.
Rule	A statement defines or constraints some aspect of the business.

4. Knowledge Memory System on Virtual Space

Knowledge memory system on virtual space has an objective view enables user to edit contents on surface by using geographical and topological correlation [18]. For this system, knowledge (contents) is visually placed based on knowledge schemas and the prospective of virtual spaces. Thence, all of them are continuously synthesized by the fusion of knowledge and filtrate by rearrange within the knowledge topology.

This mechanism knows as the Synergy knowledge mechanism that results from the spiral of knowledge development and SECI (Socialization – Externalization – Combination - Internalization) model [19].

4.1 *Knowledge Representation*

For added the acquired knowledge to the system, knowledge must be transformed into the knowledge item that represents with XML-formatting file, are called “Knowledge Card”. Knowledge card is composed of meta-knowledge and the hyperlink of information body. Meta-knowledge is the abstract of information body and hyperlink of information body obtains the detailed content of knowledge. We can acess to the information body through the hyperlink, so as to implement the integration of operations.

In knowledge card structure, it consists of three tag parts: card name, description and hyperlink of information body. Fig.3 shows a sample of knowledge card 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>
```

Fig.3 A Sample of Knowledge Card (Knowledge Instance).

4.2 *Topological of Knowledge*

On the virtual knowledge memory space, it comprised of knowledge card that has own latitude, longitude, and zooming value for determine user’s viewpoint. The geographical value of card enables user to judge the location reply on knowledge schemas and spatial clues. Fig.4 show the topological of knowledge card and filtration developing.

Notation in Figure 4:

- i — Knowledge **I**nstance from Domain knowledge.
- C — **C**oncept from Domain knowledge.
- r — Knowledge **R**ole from Inference knowledge.
- inference — Inference from Inference knowledge.
- i ⋯> r — Role of Knowledge instance in Inference step.

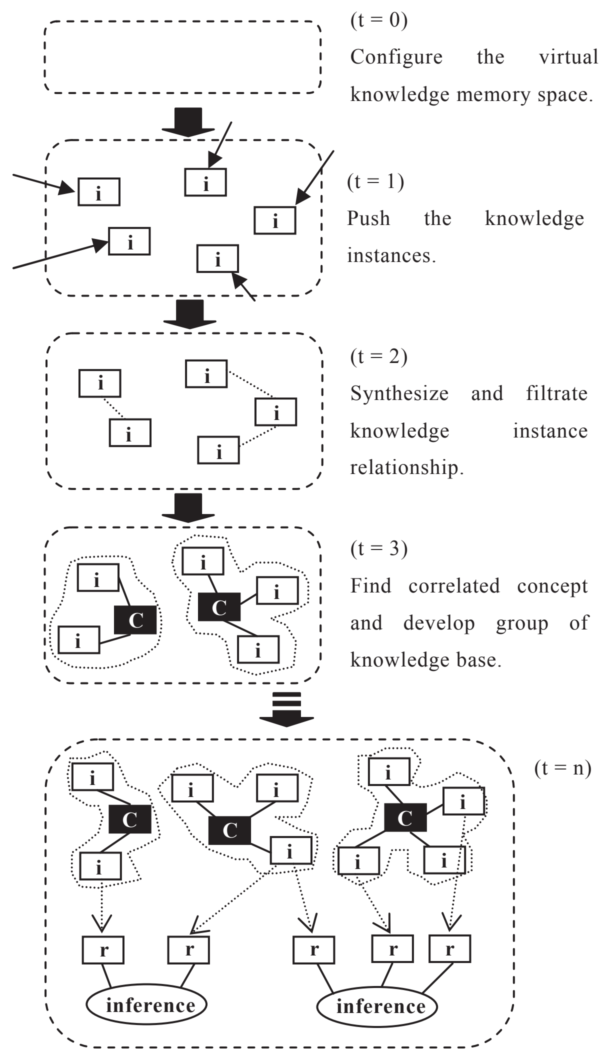

Fig.4 The Topological of Knowledge and Filtration Developing.

5. Knowledge Scenario Development Life cycle

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

To develop the knowledge scenario, user might determine the chronicle order to the inference knowledge elements, such as, Inferenec 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 the knowledge scenario by rearranging or combining old and new the inference knowledge as a life cycle. Fig.5 shows the knowledge scenario developing: Scenario (A) and Scenario (B) and their reusability.

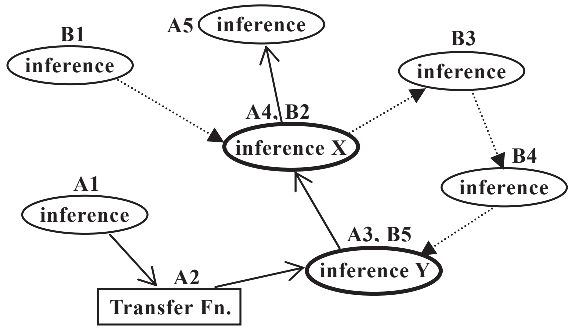

Fig.5 The Knowledge Scenario (A) and (B) and Reusability of Inference X and Y.

From Fig.5, Knowledge scenaio (A) represents with thick-line has the 5 steps (A1-A5). Knowledge scenaio (B) represents with dot-line has the 5 steps (B1-B5). Both of knowledge scenarios have two shared-inferences: Inference X and Y.

5.1 *Example*

This example describes two task knowledges that have an aim to write a basic 3D program. Task (a) is written by Java™ 3D [20] and Task (b) is written by WorldToolKit™ [21].

Task (a): Java 3D™

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).

Task (b): WorldToolKit™

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 a(1-6) and b(1-7) are the task method that similar to the knowledge scenario. Each of steps in task method equal as an inference. Some of inferences from Task (a) and (b) could shared, for example, Inference (a2) and (b1). Because of these inferences contain the knowledge instances that have same concept, for example, VirtualUniverse and WTuniverse are the knowledge instance that have the concept of creating of virtual area.

6. Conclusions

Managing knowledge through knowledge memory system is an important part of knowledge management initiative. The understanding of knowledge content and organization context is a critical success factor for the knowledge-based system development. We use the CommonKADS methodology concept to be the guideline for develop the schemas of knowledge content and organization context: Knowledge Landscape and Knowledge Atlas. We use two schemas for identify, formalized and arranging knowledge on the virtual space. On the virtual space, we represent knowledge item by Knowledge Card. Each of knowledge cards composed of latitude, longitude and zooming values that determine the user field of vision. To delivery knowledge, user creates the knowledge scenario in the series of knowledge cards. 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.

Reference

[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 2007, pp. 135-148.

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

[3] Toyoaki Nishida, “Conversational Informatic 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 2003, pp. 1070-1076.

[5] Guus Schreiber, Hans Akkermans, Anjo Anjewierden, and et al., “Knowledge Engineering and Management: The CommonKADS Methodology”, The MIT Press, 1999.

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

[7] 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., 2001.

[8] M.S.Abdullah, A.Evans, I.Benest, R.Paige, and C. Kimble, “Knowledge Modelling Using the UML Profile”, Proceedings of the 3rd IFIP Conference on Artificial Intelligence

Applications and Innovations (AIAI), Athens Greece, pp. 70 - 77.
[9] M.S.Abdullah, I.Benest, A.Evans, and C.Kimble, "Knowledge Modeling Techniques for Developing Knowledge Management Systems", the 3rd European Conference on Knowledge Management, Dublin Ireland, ISBN:0-9540488-6-5, September 2002, pp. 15-25.
[10] 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 1997, pp. 46–55.
[11] Philipe Kruchten, "Architectural Blueprints – The 4+1 View Model of Software Architecture", IEEE Software 12(6), November 1995, pp. 42-50.
[12] Heeseok Choi and Keunhyuk Yeom, "An Approach to Software Architecture Evaluation with the 4+1 View Model Archiecture", Software Engineering Conference 2002, Ninth Asia-Pacific, December 2002, pp. 286-293.
[13] Booch G., Rumbaugh J., Jaconson I., "The Unified Modeling Language User Guide", Addison Wesley, 1999.
[14] OMG Object Constraint Language 2.0 Specification, 2006.
[15] S.Boonprasert and K.Hamamoto, "CommonKADS's Knowledge Model using UML Architectural View and Extension Mechanism", the 7th International Conference on Advanced Information Management and Service, Jeju Korea, November 2011.
[16] S.Boonprasert and K.Hamamoto, "A CommonKADS's Knowledge Atlas with UML Extensions", The Joint International Symposium on Natural Language Processing and Agriculture Ontology Service 2011, Bangkok Thailand, February 2012.
[17] Mynatt E.D., Igarashi T., Edwards W.K., and Anthony LaMarca, "Flatland: New Dimensions in Office Whiteboards", ACM CHI '99, May 25-20 1999, pp. 346-353.
[18] Bederson B.B., and Hollan J.D., "Padd++: A Zooming Graphical Interface for Exploring Alternate Interface Physics ", UIST'94, 1994, pp. 17-26.
[19] S.Boonprasert, "Manage Rich Information Media in Helpdesk System with Knowledge Landscape", the International Conference on Knowledge and Smart Technologies (KST-2009), Chonburi Thailand, 2009.
[20] Dennis J Bouvier, "Getting Started with the Java 3D API", Sun Microsystems, 1999.
[21] Leno Franzen, Hans Kessock, and Dave Hinkle, "WorldToolKit Reference Manual", Engineering Animation inc., April 1999.