Chapter 3

Methodology

3.1 General

The methodology used in this research is based on the concept of knowledge engineering, which is broadly recognized in various research projects. The discipline of knowledge engineering can be applied to various areas of researches in the filed of knowledge management due to the remarkable feature of Common KADS on modeling approach, it is utilized to develop a knowledge model of SMEs.

3.2 Knowledge engineering

Knowledge engineering is presently a key technology in the knowledge society. In the past, knowledge engineering was concerned with building and developing knowledge-based systems, an objective which puts knowledge engineering in a niche of the world-wide research efforts. The discipline of knowledge engineering grew out of the early work on expert systems. Due to the popularity of knowledge-based systems, there arose also a need for systematic approach for building such systems, similar to methodologies in main-stream software engineering. [Harmelen et al., 2008] In 1990s, knowledge engineering emerged as a mature field, distinct from but closely related to software engineering. [G. Schreiber at al., 2000] [N. Milton et al., 1999] .

Traditionally, knowledge engineering was viewed as a process of "extracting" knowledge from a human expert and transferring it to the machine in a computational form. Knowledge engineering now is approached as modeling activity. [G. Schreiber et al., 1994] This modeling relates not only expert knowledge, but also the various characteristics of how that knowledge is embedded and used in the organizational environment.

Today, companies are recognizing knowledge as their key assets, which have to be exploited and protected in a fast changing, global and competitive economy. This situation has led to the application of knowledge engineering techniques in knowledge management. Knowledge engineering has become a major technique for information integration. The fast growing World Wide Web generates an ever increasing demand for more efficient knowledge exploitation and creation techniques. Knowledge engineering technologies become the key technology to solve the problem. Over the years, the discipline of knowledge engineering has evolved into the development of theory, methods and tools for developing knowledge-intensive application.

3.3 CommonKADS

In a knowledge management scenario, a knowledge modeling approach can be used to develop a model of the competence of an organization (Domingue and Montta, 2000 Although the term "knowledge modeling" can be used loosely to describe any implementation-independent model of competence, this can also be given a strict interpretation. It is referred to a precise research paradigm, which emerged in knowledge engineering as a reaction to the mining view of knowledge acquisition (Kidd, 1987).

One of the salient features of CommonKADS is the modeling approach. The templates for each of the models form the core of the methodology (G. Schreiber et al., 1994). The knowledge modeling approach in earlier versions of KADS has matured in CommonKADS, and nowadays includes extensive facilities for modeling of domain knowledge. The CommonKADS expertise model supports the introduction of ontologies as a mechanism for generating and vocabulary of the domain knowledge.

The CommonKADS methodology has evolved over almost two decades. It was originally conceived as the KADS approach to knowledge acquisition [Wieliinga and Breuker, 1984]. It has been extended, adapted and revises over the years until its most recent formulation by Schreiber and his teams (G. Schreiber et al., 2000). It is

also emphasized that the role of a methodology based on the knowledge modeling paradigm is to provide a suite of techniques to support knowledge analysis in an organization, in a wide range of scenarios.

The separation of those three view points comprising data, functional and control view is similar with those identified by current software engineering approach. Method employed in CommonKADS differs in several ways from conventional modeling techniques. One of the major differences is the nature of the link between domain knowledge ("data") and the inferences ("function").

CommonKADS methodology offers a structured approach to break down and structure knowledge engineering process. CommonKADS model suite for creating requirements specification for knowledge system is shown in Figure 3.1 (G. Schreiber at al., 2000) below. The method enables a top-down approach and provided handles for quality control and feasibility assessment.

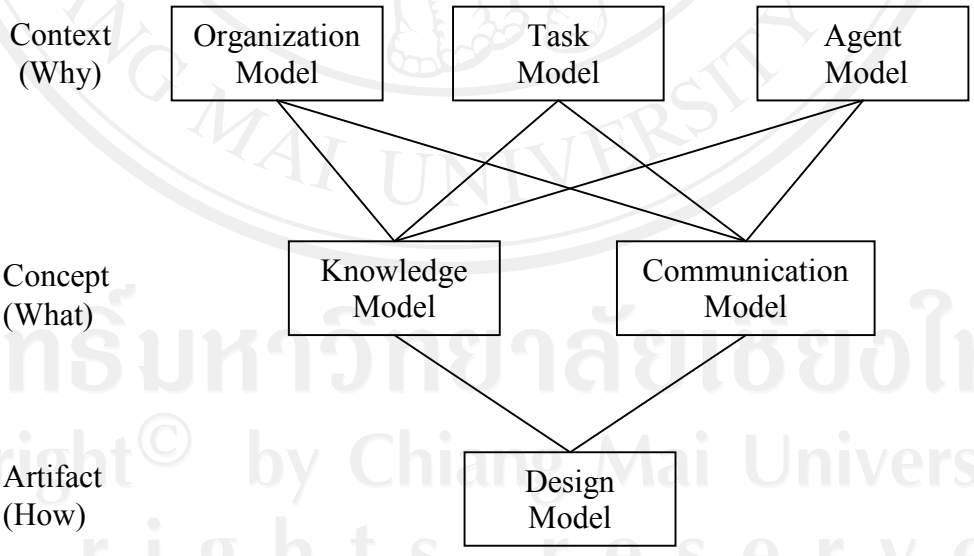

Figure 3.1: The CommonKADS model suite

- **Context level** analyzes the organizational environment and the corresponding critical success factors for a knowledge system.
  - *Organization model* supports the analysis of the major features of an organization, in order to discover problems and opportunities for the knowledge system establish their feasibility, and assess the impacts on the organization of intended knowledge actions.
  - *Task model* 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 competence skills.
  - *Agent model* are the agents who are the executors of a task. It 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 the task.
- **Concept level** yields the conceptual description of problem solving function and data that were handed and delivered by knowledge system.
  - Knowledge model explicates in detail the type and structures of the knowledge used in performing a task. It provides an implementation-independent description of the role that different knowledge components play in problem-solving, in a way that is understandable by humans.
  - Communication model shows transaction between the agents involved in a conceptual and implementation-independent way, just as with the knowledge model.
- **Artifact level** combines the above levels together in the design model in order to construct the requirement specification for the knowledge system.
  - Design model gives the technical system specifications in term of architecture, implementation platforms, software modules, representational constructs, and computational mechanism needed to implement the functions used in the knowledge and communication models.

In terms of knowledge processes, CommonKADS methodology provides knowledge templates for supporting the knowledge modeling process, which constitute predefined reusable knowledge models and which have proven to work in the past, the details of knowledge templates will describe in the next topics.

This methodology concerns all knowledge activities i.e. create, store/retrieve, share, and representation. However, knowledge sharing methods did get much attention in this methodology. The notions of network and organization culture are analyzed in organizational model in the context level. Although, this method did not provide a concrete process for handling a specific task, it is detailed enough to be able to apply in any knowledge intensive task. This methodology is recommended for KM projects that concern the knowledge exchange between agents in intra- and inter-organization.

There are principally 3 categories of question which can be presented in the model suite as follows:-

- Why? Why is a knowledge system a potential help or solution? For which problems? Which benefits, costs, and organizational impacts does it have? Understanding the organizational context and environment is the most important issue here
- What? What is the nature and structure of the knowledge involved? What is the nature and structure of the corresponding communication? The conceptual description of the knowledge applied in a task is the main issue here
- How? How must the knowledge be implementing in a computer system? How do the software architecture and the computational mechanisms look? The technical aspects of the computer realization are the main focus here.

Accordingly, the knowledge engineering divides the structure of knowledge according to cognition behaviors into 3 levels, namely task knowledge, inference knowledge, and domain knowledge. Each level has the details as follows:

1. The task knowledge is the knowledge about the objective of the considered task.
2. The inference knowledge is the knowledge about the steps in solution.
3. The domain knowledge is the knowledge of the principles, how-to, or causes and effects involved in the task of interest.

Figure 3.2: Task knowledge representation.

Figure 3.3: Inference knowledge representation.

Figure 3.4: Domain knowledge representation.

Knowledge model can be partially reused in new application by using an available common task template. Task templates form a common type of a reusable combination of model elements. A task template supplies knowledge engineer with inferences and tasks that are typical for solving problem of a particular type. In

addition, task templates can be used by the knowledge engineer as a template for new application.

CommonKADS provides this common knowledge model template which distinguishes into two groups of task type, which are *''analytic'* task and *'synthetic'* tasks. These are further subdivided into a number of task type based on the type of problem tackled by the task. In analysis task, system is preexists but not completely known.

All analytic tasks take as input data about the system and produce some characterization of the system as output. In analytic tasks the system preexists although it is typically not completely known.

For synthetic task, in contrast, the system does not exist. The purpose of the task is to construct a system description. The input of a synthetic task typically consists of requirements that the constructed system should satisfy. An overview of the main features of these knowledge model templates are shown in Table 3.1 (G. Schreiber et al., 2000).

Table 3.1: Overview of Task Types

Task type	Input	Output	Knowledge
Analysis	*System observations*	*System characterization*	*System model*
Classification	Object features	Object class	Feature-class associations
Diagnosis	Symptoms/ complaints	Fault category	Model of system behavior
Assessment	Case description	Decision class	Criteria, norms
Monitoring	System data	Discrepancy class	Normal system behavior
Prediction	System data	System state	Model of system behavior
Synthesis	*Requirements*	*System structure*	*Element, constraints, preferences*
Design	Requirements	Artifact description	Components, constraints, preferences
Configuration design	Requirements	Artifact description	Components, skeletal designs, constraints, preferences
Assignment	Two object sets, requirements	Mapping set 1 → set 2	Constraints, preferences
Planning	Goals, requirements	Action plan	Actions, constraints, preferences

Table 3.1: Overview of Task Types (Cont.)

Task type	Input	Output	Knowledge
Scheduling	Job activities, resources, time slot, requirement	Schedule = activities allocated to time slots of resources	Constraints, preferences
Modeling	Requirements	Model	Model element, template models, constraints, preferences

CommonKADS is a good candidate for becoming the de facto European standard and point of reference for knowledge engineering. (G. Schreiber et al., 1994). As a framework, CommonKADS has given rise to numerous research projects.