Chapter 3
Knowledge Management and Engineering in Power Delivery Business

Everyone would agree that knowledge is the most important ingredient of corporate strategies to obtain a sustainable competitive advantage in the current economy. Many organizations have put intensive investment to acquire expertise for its employees, while the potential to lose these expertises is very fragile. As the capital is allowed to move freely within industry, the staff turnover is also a big issue in this modern economy. When the employees leave, they take with them all valuable knowledge, leaving behind the losses to the company, not only the non-return on personnel investment but also the competitive position. And again company has to re-spend times and money to recover those expertise and competitive position back to the company. Knowledge retention needs to be put in place to capture such valuable knowledge. Nowadays many organizations employ knowledge management discipline to manage its organizational knowledge in order to develop, retain and reuse such knowledge in order that its business success could be fully secured.

Knowledge Management (KM) facilitates the capture, storage, and dissemination of knowledge using information technology and management approaches. This would allow the organization to fully utilize its knowledge assets. The main component in knowledge management system is knowledge base. It thus needs some means to create knowledge based system. Knowledge Engineering (KE) provides method and methodology to design and construct knowledge systems. It covers capturing, analyzing, validating and modeling a domain of knowledge. There are several knowledge engineering approaches have been developed during the last decades. A well-known one includes the CommonKADS which offers a guideline for design and implementation of knowledge systems. CommonKADS has been broadly applied in power business, for instances, knowledge management for planning, operation, maintenance, pricing negotiation, asset management and regulatory issues.

In the field of power distribution asset management, a decision to be made on infrastructure investment needs to be cautiously made and requires the involvement of human experts. But the availability of experts is somehow scarce or difficult to retract them from their overloaded work. On the other hand, it would make investment decision efficiently and effectively, if there is a decision support system (DSS) available to assist the decision making process. It is the intention of the thesis to develop such decision support tool. The proposed DSS emulates the decision making process of human experts; thus the knowledge required for assisting making a decision are to be obtained from utility managers, engineers and technicians. Knowledge management and engineering facilitates the knowledge elicitation, analysis, modeling to obtain the essential knowledge used in assembling the DSS. Particularly, when interacting with human experts to capture the knowledge they possess, KM&E provides a set of useful guidelines for such interaction. KE methodologies, particularly CommonKADS Classification and Assessment task templates, help the knowledge engineer to decompose the application task to the level where they can identify the knowledge components that can be used in decision making process. These knowledge components are then modeled and categorized using ontology development methodology. These ontologies accordingly represent the classes, properties, and relationships of all relevant distribution network assets; the collection of asset instances would act as the database of the DSS. This chapter describes the knowledge elicitation, analysis and modeling methodologies that are applied in this research study.

3.1 Chapter Overview

In this chapter, the knowledge and knowledge management is first introduced. Then the principle and methodologies of knowledge engineering are discussed; followed by the description of knowledge capturing approaches. The techniques for modeling knowledge especially on power distribution system are presented in the later section.

3.2 Knowledge and Knowledge Management

In the world of economy of speed, everything needs to be done at fast; products have to be entered the market fast; customers' requests need to be responded quickly; organization cannot afford the costs and times for error anymore. Organization has to utilize every resource it has to satisfy the customer's expectations; so starting from scratch is not an option. Actually organizations have already had experts of diverse fields in house. The issue is how to retain and utilize these knowledge in order to gain the maximum benefit. This is why knowledge management comes into play.

Before attempting to address the question of knowledge management, it's probably appropriate to develop some perspective regarding the literal term called knowledge, which there seems to be such a desire to manage. Figure 3.1 provides a very good illustration of what seems to be called *knowledge*.

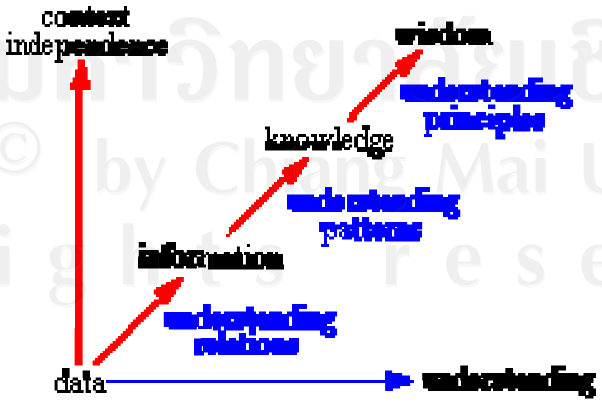

Figure 3.1 Evolution of knowledge

Data, information and knowledge are three often-encountered words that belong closely together, seem to have slightly different meanings, yet are often used interchangeably as synonyms [38]. However, in the context of knowledge modeling, they are quite different in the meanings. Data is an uninterpreted signal that has been come across with the sensors, either human sensory organs such as eyes, ears, or noses, or hardware sensors like thermometer, clock, or measuring devices which used in people's daily living. Data is raw; simply exists; could be any forms, usable or not. It does not have meaning of itself. A red, green, or yellow traffic light is one example. When these data are compiled to obtain some meaning, it becomes information. The meaning can be useful, but does not have to be. Red traffic light, for example, gives information as indication to stop to the car driver, but it means nothing to the caveman who happens to show himself in the middle of Bangkok in rush hour. Even though data is the same, but the information is not. Knowledge is the information for action, such that its intent is to be useful. When it comes to be a practical action to achieve the purpose, people use knowledge to perform. When the driver sees the red light, he is informed that he must stop the car, the knowledge he uses to perform his action are to release the accelerator and press the brake to make the car stop. Collecting the data is straightforward; it can be done through reading of sensors or measuring devices. Interpretation of data to obtain information may require an understanding of the subject or some training. But to possess the knowledge, people need workplace learnings and job experiences. In decision making process, data needs to be collected and stored; analyzed it together with other data to obtain information; and finally the decision maker uses his knowledge to take an action. However, most of the commercial DSS claim that they are capable of performing the decision making without an involvement of human experts; but in reality the human expert has to get involved somehow. So, the question to be answered here is not about getting the ideal DSS, rather it is about providing as much possible as the information to the decision make, as well as the replication of the basic problem solving process of human experts.

It is obvious that people use knowledge to solve the problems they have encountered. The level of knowledge usage depends on the complexity of problems and who is going to use it. Peter Drucker [39], the well-known philosopher, coined the term *knowledge work* by characterizing the *knowledge workers* as those who have high levels of education and specialist skills and combined with the ability to apply these skills to identify and solve problems. In this context, managing knowledge is like managing people who have such knowledge.

There seems to be several brilliant definitions of knowledge management (KM) given by philosophers, but the meaning given by Brian Newman [40] seems to fit to this research study. Newman proposed the general knowledge model for organizing knowledge flows into four primary activity areas: knowledge creation, retention, transfer and utilization. The graphical representation of this model is shown in figure 3.2.

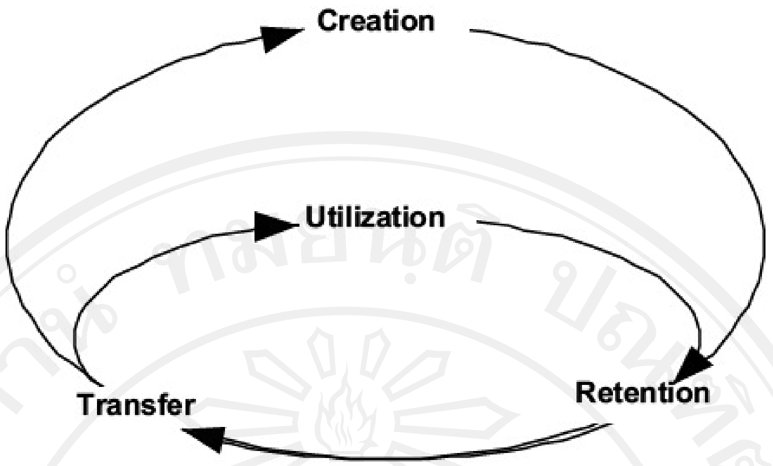

Figure 3.2 The general knowledge management model [40]

Here are the descriptions of activities that take place in the general knowledge management model circle.

Knowledge Creation: this comprises activities associated with the entry of new knowledge into the system, and includes knowledge development, discovery and capture.
Knowledge Retention: this includes all activities that preserve knowledge and allow it to remain in the system once introduced. It also includes those activities that maintain the viability of knowledge within the system.
Knowledge Transfer: this refers to activities associated with the flow of knowledge from one party to another. This includes communication, translation, conversion, filtering and rendering.
Knowledge Utilization: this includes the activities and events connected with the application of knowledge to business processes.

There are also many literatures that research deeply into the field of knowledge management. The most well known and cited ones are the books of David A. Garvin, Ikujiro Nonaka and Hirotaka Takauchi, and Peter M. Senge. These books provide the fundamental for creating, retaining, transferring, measuring and utilizing knowledge. As Bhagavad-Gita says “the wise see knowledge and action is one”, knowledge cannot be formulated by just seeing, listening, studying or training; it needs to incorporate the aforementioned with action. Garvin [41] has introduced the learning process of people to acquire the knowledge, learn how to learn on the other hand. He outlines the processes of acquiring the knowledge and emphasizes the importance of team leadership. Argyris [42] proposes double loop learning theory which pertains to learning to change underlying values and assumptions. An important aspect of the theory is the distinction between an individual's “espoused theory” (what they would like others to think they do) and their "theory-in-use" (what they actually do); bringing these two into congruence is a primary concern of double loop learning. The focus of the theory is on solving problems that are complex and ill-structured and which change as problem-solving advances. The end result of double loop learning should be increased effectiveness in decision-making and better acceptance of failures and mistakes. Senge [43] emphasizes the role of human components of the company, people don't just cultivate their limited territory and privileges, but where they have to work together and take responsibility for their shared future, working on creating maximum synergy and maximum ability to deal with the whole situation. As stated earlier, human is the crucial ingredients in cultivating KM and this book lays down the foundation of human role and how to tap and exploit the knowledge resided in the people, hence the success of KM implementation. Ikujiro Nonaka [44] proposes the means to manage both explicit and tacit knowledge owned by the company. SECI model provides the mechanism to tap organizational knowledge and embeds them in processes and products of the company. He also advises the enabling conditions and five-phase model of the organizational knowledge creation process. And finally, when implementing knowledge management managers shall employ it throughout the company, not individual division. Finally, Edvinsson [45] considers intellectual capital primarily as the hidden values constituting the gap between market value and book value. To leverage these assets, organization needs to know what the staffs know that gives the organization a competitive advantage relative to others in marketplace. He proposed the framework to structures, packages, and measure intellectual capital as a guide for management and knowledge worker activity. It also helps orient the setting of objectives and guide the performance of knowledge workers in terms of five focus area: financial, customer, human, process, and renewal/development. Table 3.1 provides a summary on the knowledge management framework given by above knowledge management gurus.

Table 3.1 Knowledge management framework

Framework	Concept	Feature	Organization
Double-Loop Learning (Argyris, 1973)	Balance How with Why	Prevent human Error from Defensive Routine	Teaching the Smart People How to Learn
The Fifth Discipline (Senge, 1990)	System Dynamics for Changes	Prevent Snow Balling, Balance Operating Parameters	Develop learning skill during Changes
Knowledge Creating Company (Nonaka, 1995)	Knowledge Creation Process	Create Innovation	Knowledge production
Intellectual Capital (Edvinsson, 1997)	IC measurement	Market value control	High IC public company
Learning in Action (Garvin, 2000)	Action to Learning Opportunity	Develop learning skill at work	Critical tasks or core business line

Since knowledge is critical asset to company, only the company that can efficiently manage this critical asset can thrive the success. Furthermore only the company that understands its internal environment and choose the right framework that suits its environment can manage its knowledge asset successfully.

3.3 Knowledge Engineering

While knowledge management facilitates the knowledge creation, retention, transfer and utilization knowledge using both information technology and management disciplines; knowledge engineering provides method and methodology to design and construct knowledge systems. It covers capturing, analyzing, modeling, validating and storing a domain of knowledge. Knowledge engineering is the process of codifying an expert's knowledge and experience in a form that can be stored, accessed and processed by a computer system. It helps organization transform tacit knowledge resides in people's head into an explicit form in order that it can be stored and distributed to the users.

Knowledge Engineering was originally set up as a new discipline in artificial intelligence (AI) with the objective of providing methods and tools for constructing knowledge based systems in a systematic and controllable way [46, 47]. It involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise. At present, knowledge engineering refers to the building, maintaining and development of knowledge-based systems [48, 49].

The core element in knowledge engineering process is concerned with how to acquire the knowledge. Over years knowledge engineers have developed a number of principle, methods and tools that have made knowledge acquisition an efficient and effective activity. The principles having been laid out for such efficient and effective acquisition rely on the followings [50].

- Recognize that there different types of knowledge; e.g. declarative and procedural knowledge, static and dynamic knowledge, tacit and explicit knowledge, or abstract and specific knowledge.
- Recognize that there different types of experts and expertise, depending on their trainings and experiences.
- Recognize that there are different ways of representing knowledge.
- Recognize that there are different ways of using knowledge.
- Use structured methods.

3.3.1 Knowledge Elicitation

Knowledge elicitation is the process of getting the data needed for the knowledge modeling. There are a number of elicitation techniques such interviewing, protocol analysis, laddering, concept sorting and repertory grid [51]. The application of each technique depends on the types of knowledge, the types of expertise, or the ways of knowledge representation having been just mentioned above. However, the most commonly used one is an interviewing. By interviewing, both elicitor and knowledge provider can naturally and freely approach and interact each other.

Unstructured interview has no agenda set either by interviewer or expert, but does not mean that they have no goal for interview. Goal is clearly defined among both but a detail for conducting elicitation is made up to situation. Structured interview, on the other hand, has the advantage that it provides structured transcripts that are easier to analyze and control interviewing time but it might prevent experts revealing their knowledge freely. Experts do not work through a particular scenario extracted from domain by the interviewer, rather the experts generates their own scenarios as the interview progresses, so a compromised interviewing technique, a semi-structured, would be able to fill the gaps caused by aforementioneds.

An effective way to perform the knowledge elicitation using interview is to start with an unstructured interview to allow experts expressing freely on how they solve the problem, e.g. asset categorization task or risk assessment task, etc. After a general outline of task is described and some key concepts are identified, a structured interview will be conducted to fill the gaps that exist and to induce another concept. In addition, the structured interview also assists the elicitator in obtaining information about key concepts and their relations. During the structured interview, it is however helpful to constrain the elicitation's interview to a question set of probes, each with specific function. This is to prevent the interview from broadening to irrelevances. Table 3.2 contains a list of probing questions which was applied to get insightful information.

Table 3.2 Probes to elicit information in structured interviews [38].

Probe Code	**Question template**	**Effect**
P1	*Why would you do that?*	Converts an assertion into a rule
P2	*How would you do that?*	Generates lower-order rules
P3	*When would you do that?* *Is<the rule>always the case?*	Reveals the generality of the rule and may generate other rules
P4	*What alternatives to <the prescribed action/decision> are there?*	Generates more rules
P5	*What if it were not the case that <currently true condition>?*	Generates rules for when current condition does not apply
P6	*Can you tell me more about <any subject already mentioned>?*	Used to generate further dialogue if expert dries up

There is also a pitfall of an interview that expert will only tell what they can verbalize. If there are non-verbalized aspects in the domain, that is the expert knows how to solve the problem but is unable to explain it, the interviewer will not cover them. To remedy this problem, the technique of protocol analysis can be supplemented to an interview. The expert is asked to carry out his real job and verbalize all his/her thought while processing the task. The protocol is then later analyzed to identify the structure of reasoning process and knowledge components.

In this research, the main source of knowledge is utility engineers and technicians; the elicitation techniques employed to capture the knowledge on power distribution asset management are primarily based on an interview and a protocol analysis. The staff from diverse department, e.g. planning, design, installation, operation and maintenance of distribution facilities are involved in the study.

3.3.2 Knowledge Engineering Concept

The conceptual foundation of knowledge engineering lies in the modeling framework for knowledge representation. In the past, knowledge engineering was regarded as a transferring of knowledge from an expert into a knowledge base. Approaches based on this viewpoint often failed. The reasons were mentioned just before that there are different types of knowledge and expertise; the experts are often unaware of experiences they use to solve their problems. Hence, crucial pieces of knowledge are not directly accessible and need to be constructed and structured during the knowledge acquisition phase. In the viewpoint of codifying an expert's knowledge and experience into manageable form, the knowledge acquisition process in this context is therefore a model construction process. On the other hand, the construction of a knowledge-based system means building a computer model with the aim of realizing problem-solving capabilities that are comparable to a domain expert.

There are two major notions developed in knowledge engineering community: the notion of problem solving methods (PSM) and ontologies [52]. These two concepts provide the backbone for building structured and reusable knowledge models. Ontologies are concerned with static domain knowledge whereas problem solving methods deal with dynamic reasoning knowledge [53]. PSMs and ontologies can be seen as complementary knowledge components to construct knowledge systems.

The notions of PSMs and ontologies have been reflected in various modeling frameworks such as CommonKADS, MIKE, and PROTÉGÉ. Further details of these two notions are discussed below.

3.3.2.1 Problem-Solving Methods

Problem solving methods (PSM) are nowadays recognized as valuable components for constructing knowledge based systems (KBS). The notion of PSM is present in leading knowledge engineering frameworks, CommonKADS for example. PSMs describe the reasoning process of a KBS in an implementation- and domain-independent manner. A PSM defines a way of how to achieve the goal of a task. It has inputs and outputs and may decompose a task into subtasks. In addition, a PSM specifies the data flow between its subtasks. Control knowledge determines the execution order and iterations of the subtasks of a PSM.

A PSM may be characterized as follows:

- A PSM specifies which inference steps have to be carried out for achieving the goal of a task.
- A PSM defines one or more control structures over these steps.

- Knowledge roles specify the role that domain knowledge plays in each inference step. These knowledge roles define a domain-independent generic terminology. There are two types of roles: static roles describe the domain knowledge needed by the PSM; dynamic roles form the input and output of inference steps.

How to develop a problem solving methods is a principal question that knowledge community tries answer. One way to do this is by analyzing human problem-solving behavior and representing this behavior computationally. Another way to do this is to perform reverse engineering of existing expert systems. These two ways of developing PSMs essentially involve a creative activity, for which no methodological support exists.

3.3.2.2 Ontologies

Ontologies have become a popular research topic and have been investigated by several Artificial Intelligence research communities, including knowledge engineering, natural-language processing and knowledge representation. Ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups. They provide a common vocabulary of an area and define the meaning of the terms and the relations between them. Ontologies are usually organized in taxonomies and typically contain modeling primitives such as classes, relations, functions, axioms and instances [54]. Popular applications of ontologies include knowledge management, natural language generation, enterprise modeling, knowledge-based systems, ontology-based brokers, and interoperability between systems [53].

Basically, the role of ontologies in the knowledge engineering process is to facilitate the construction of a domain model. In the last decades a wide range of methodologies for building ontologies have been proposed. Apart from minimal differences related to domain and application constraints, ontology engineering methodologies usually introduce an iterative process consisting of the following phases [55]:

- Domain analysis (including requirements analysis and knowledge acquisition);
- Conceptualization of the domain knowledge;
- Ontology implementation,
- Ontology population (i.e. generation or integration of ontology instances),
- Ontology evaluation,
- Ontology refinement, and
- Ontology maintenance.

All the ontology development programs more or less follow this generic framework in building ontologies.

3.4 Application of Knowledge Engineering Methodologies in Power Distribution System

This section is aimed to discuss the application of knowledge engineering particularly knowledge modeling in the field of power distribution system. Only two methodologies: CommonKADS and Ontology101 are focused in this thesis.

3.4.1 CommonKADS Template Knowledge Model

The CommonKADS knowledge model [38] is a tool that helps to clarify the structure of a knowledge intensive business task. The knowledge model provides a specification of the data and knowledge structures required for the application. The model is developed as part of the analysis process. It is therefore phrased in the vocabulary of the application, meaning both the domain and the reasoning task. A knowledge model composes of three categories; each capturing a related group of knowledge structures and can be described as follows.

- *Domain Knowledge* specifies the domain-specific static information and knowledge objects that involved in an application, for example cable, joint, and termination.
- *Inference Knowledge* describes how domain knowledge can be used to carry out a reasoning process, describes the basic inference steps on how to make use of the domain knowledge.
- *Task Knowledge* describes what goals and application pursue (e.g. classification, diagnosis, assessment), and how these goals can be realized through a decomposition into subtasks and inference ultimately.

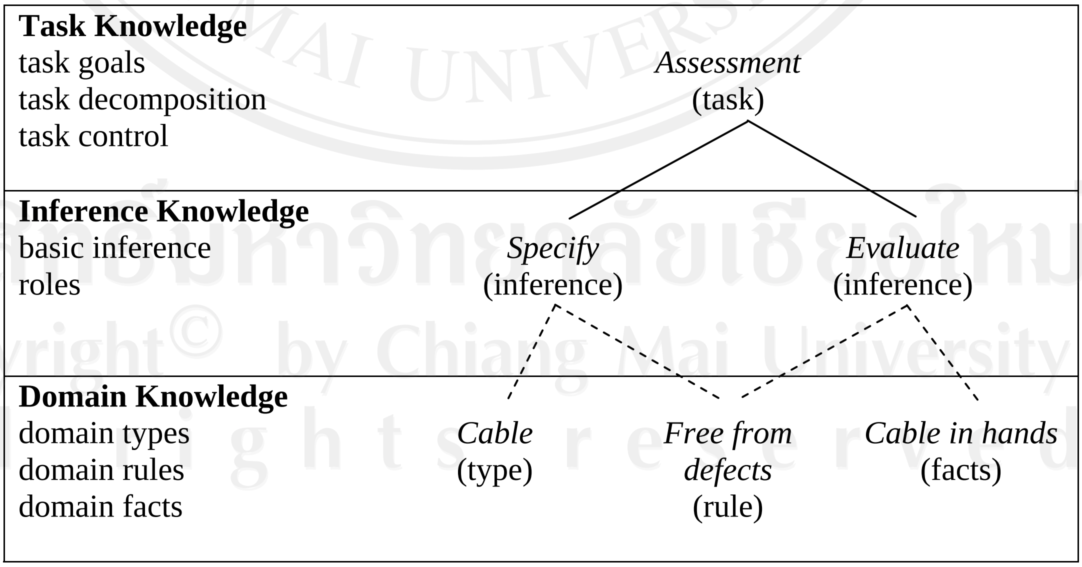

Figure 3.3 Overview of knowledge categories in the knowledge model.

Figure 3.3 shows an overview of knowledge categories in the knowledge model. At the right shows the example of the knowledge elements in installation workmanship assessment of underground cable.

To enable knowledge modeling process, the CommonKADS has provided a knowledge engineer with a collection of predefined sets of model elements that can be used for modeling a relatively simple knowledge-intensive task, i.e. task template. A task template supplies the knowledge engineer with inferences and tasks that are typical for solving a problem of a particular type. As shown in figure 3.4, the task types can be distinguished into two groups: analytic task and synthetic task.

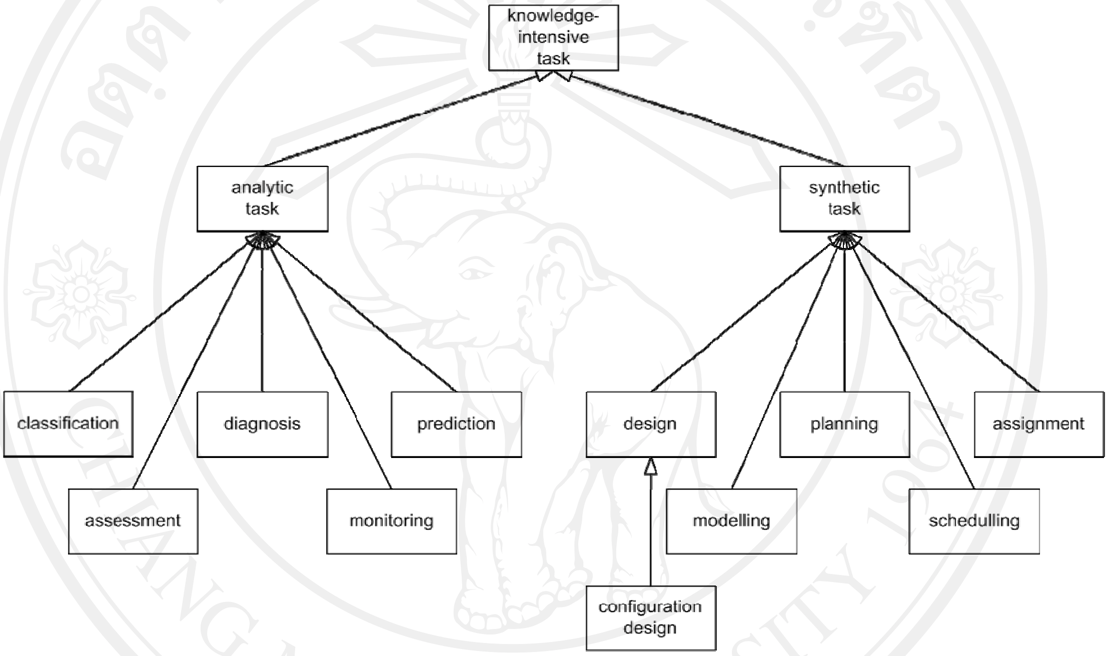

Figure 3.4 Hierarchy of knowledge-intensive task type based on the type of problem being solved [38]

Analytic task deals with the systems that have already existed although it is not typically completely known. All analytic tasks take as input some data about system and produce some characterization of the system as output. For the synthetic tasks, on the other hand, the system does not yet exist. The purpose of this task is to construct a system description. The input of a synthetic task typically consists of the requirements that the system to be constructed should satisfy.

Analytic and synthetic tasks are further subdivided into a number of task types. This subdivision of tasks is based on the type of problem tackled by the task. For example, a diagnosis task is concerned with finding a malfunction that causes unusual system behavior. Table 3.3 and 3.4 provide an overview of the main features of analytic and synthetic task types.

Table 3.3 Overview of analytic task [38]

Task type	**Input**	**Output**	**Knowledge**	**Features**
Analysis	System observations	System characterization	System model	System description is given.
Classification	Object features	Object class	Feature-class associations	Set of classes is predefined.
Diagnosis	Symptoms/ complaints	Fault category	Model of system behavior	From output varies (casual chain, state, component) and depends on use made of it (troubleshooting).
Assessment	Case description	Decision class	Criteria, norms	Assessment is performed at one particular point in time (cf. monitoring).
Monitoring	System data	Discrepancy class	Normal system behavior	Systems changes over time. Task is carried out repeatedly.
Prediction	System data	System state	Model of system behavior	Output state is a system description at some future point in time.

Table 3.4 Overview of synthetic task [38]

Task type	**Input**	**Output**	**Knowledge**	**Features**
Synthesis	Requirements	System structure	Element, constraints, preferences	System description needs to be generated.
Design	Requirements	Artifact description	Components, constraints, preferences	May include creative design of components.
Configuration design	Requirements	Artifact description	Components, skeletal designs, constraints, preferences	Subtype of design in which all components are predefined.
Assignment	Two object sets, requirements	Mapping set 1 → set 2	Constraints, preferences	Mapping need not be one-to-one
Planning	Goals, requirements	Action plan	Actions, constraints, preferences	Actions are (partially) orders in time.
Scheduling	Job activities, resources, time slot, requirement	Schedule = activities allocated to time slots of resources	Constraints, preferences	Time-oriented character distinguishes it from assignment.
Modeling	Requirements	Model	Model element, template models, constraints, preferences	May include creative "synthesis"

This research study deals with the categorization of power distribution network asset as well as the assessment of asset performance to assist the investment decision making. Hence, two templates of CommonKADS knowledge model are employed to assist an extraction of the key knowledge concerning the distribution of electric power from utility substation to its customers. The following sections provide the explanation of these two approaches as well as their example applications applied in the power distribution field.

Classification Knowledge Model

Classification is regarded as the most basic modeling method that human applies for learning things that they have encountered. Human classifies one thing from the others by differentiating their features or characteristics. The classification of plants or animals is the prototypical of this type. In classification, an object needs to be characterized in terms of the class it belongs. The underlying knowledge typically provides for each class constraints on the values of object features [38]. Although classification usually involves with *natural* objects, the concept can be applied to any classification jobs.

Table 3.5 General characterization of classification template

Goal:	Establishing the correct class (or category) for an object. The *object* is hence made available for an inspection.
Typical example	Classification of distribution feeders, classification of circuit breaker
Terminology	
Object	The object of which one wants to find the class or category, e.g. feeder no. PI-411.
Class	A group of objects that share similar characteristics, e.g. overhead feeder
Attributes	A characteristic that can either be observed or inferred, e.g. insulation.
feature	An attribute-value pair that holds for certain object, e.g. PE insulation
Inference	
Generate	Generate the possible candidate class of considered object
Specify	Specify the attribute that the class holds
Match	Match the value of each attribute that the object holds to those that characterize the class
Input	The object of which the class need to be established
Output	The class that object belongs

The general characterization of classification modeling can be described by table 3.5 above whereas the corresponding inference structure is shown in figure 3.5 below.

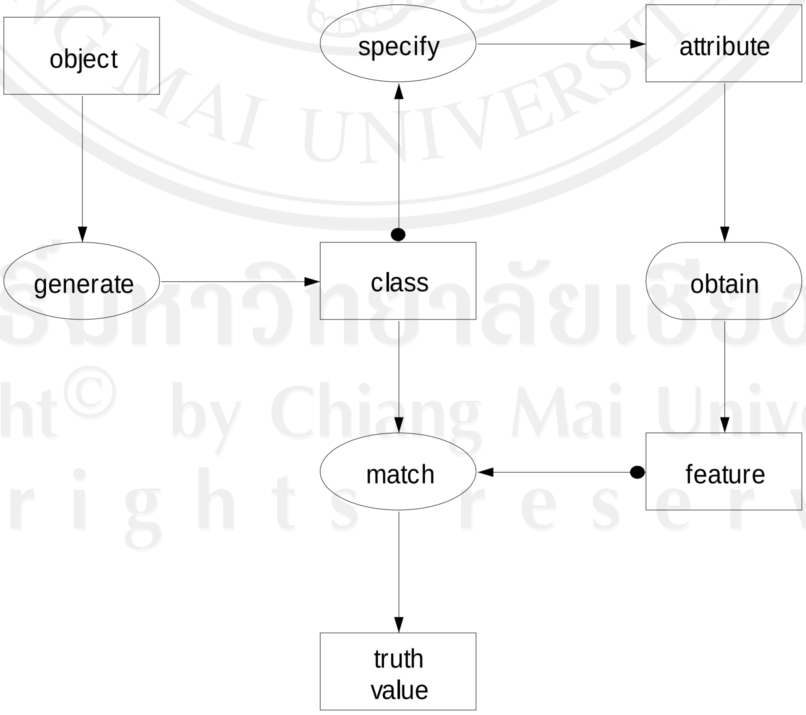

Figure 3.5 Classification task template [38]

Following is a practical example to explain how this classification template works (see figure 3.5). If the correct *class* of an electric cable *object* is to be established, a full set of candidate solutions is first *generated*, e.g. all potential electric cable class. Since a set of candidate solutions is defined first, this approach is called solution-driven. From the above mentioned example, after gathering all related information particularly about electric cable, two classes, i.e. overhead cable (OH) and underground cable (UG), are defined. Then the characteristics of interest are *specified*. For this case, the domains of interest are the current carrying and voltage withstand capability of the cable; the *attributes* that define such characteristics are defined as indicated as shown in table 3.6. Once the *feature* (an attribute-value pair) of a certain object is *obtained*, then this object can be *matched* to the class it belongs. At the end, it can be concluded that a piece of electric cable of this example belongs to an overhead cable.

Table 3.6 Classes and object of power distribution cable

Class: *OH cable*		**Class:** *UG cable*		**Object:** *Electric cable*	
Attribute:	**Typical value**	**Attribute:**	**Typical value**	**Attribute:**	**Value**
Conductor Material:	Cu., Al.	Conductor Material:	Cu., Al.	Conductor Material:	Al.
Conductor X-section	mm^2	Conductor X-section	mm^2	Conductor X-section	185 mm^2
Insulation material	XLPE, PVC	Insulation material	XLPE, paper	Insulation material	XLPE
Ampacity	amp	Shield wire material:	Cu., Al.	Shield wire material:	none
Length	m	Shield wire X-section	mm^2	Shield wire X-section	none
		Jacket material	PE, lead	Jacket material	none
		Ampacity	amp	Ampacity	400 amp
		Length	m	Length	500 m

By using the classification knowledge model, all the components used in power distribution network can be categorized into different classes depending on their functionality and performance. The details of class-attribute development, however, will be elaborated in the later chapter where the power distribution asset categorization is focused.

Assessment Knowledge Model

Another CommonKADS knowledge model employed in this research is an *assessment* template. The assessment template is employed to assist the expert's knowledge capture on distribution network facility performance evaluation. The same way as proposed by *classification* template, the assessment knowledge model offers a guideline for interview, analyze, model and utilize the captured knowledge. It helps knowledge engineer to decompose the application task to the level where they can identify the knowledge components that can be used in decision making process. The goal of assessment in general is to take a decision based on data about a case and a set of norms or criteria. The general characterization of the template and the corresponding inference structure is shown in table 3.7 and figure 3.6 respectively

Table 3.7 General characterization of assessment template

Goal:	Find a decision category for a case based on a set of domain specified norms
Typical example	Decide whether electric cable is still in a good condition
Terminology	
Case	The case to be assessed, e.g. data of electric cable
Decision category	e.g. good, poor, failed
Norms	Domain knowledge that is used in making the decision, e.g. criteria that indicate the performance of electric cable
Inference	
Abstract	Abstract the related cases that will be brought into decision and discard the rest
Specify	Generate a lists of norms that could be evaluated for a certain case
Select	Selects norm from the list for further evaluation
Evaluate	Evaluate a particular norm for a case at hand against a set of criteria and returns the truth value
Match	Take all results of norms evaluation as input to determine whether a decision is reached
Input	Data about the case, case-specific norms
Output	A decision category

As shown in figure 3.6, five inferences are required for assessment task. The *abstract* inference takes some case data as an input and produces a new abstracted case datum as a result. Only related case data will be abstracted for further analysis, irrelevances will be scraped off. The *specify* inference generates a lists of norms that could be evaluated for a certain case. Norm prescribes what ought to happen. The *select* inference selects one norm from the list. The selection can be done randomly, based on heuristics like “first the most likely one to fail.” The *evaluate* inference evaluates a particular norm for a case at hand against a set of criteria, and returns the truth value, indicating whether the norm hold for the case. Finally, the *match* inference takes as input all results of norms evaluation, and succeeds if a decision can be reached. For example, the decision can be reached if all norms have been evaluated and are true.

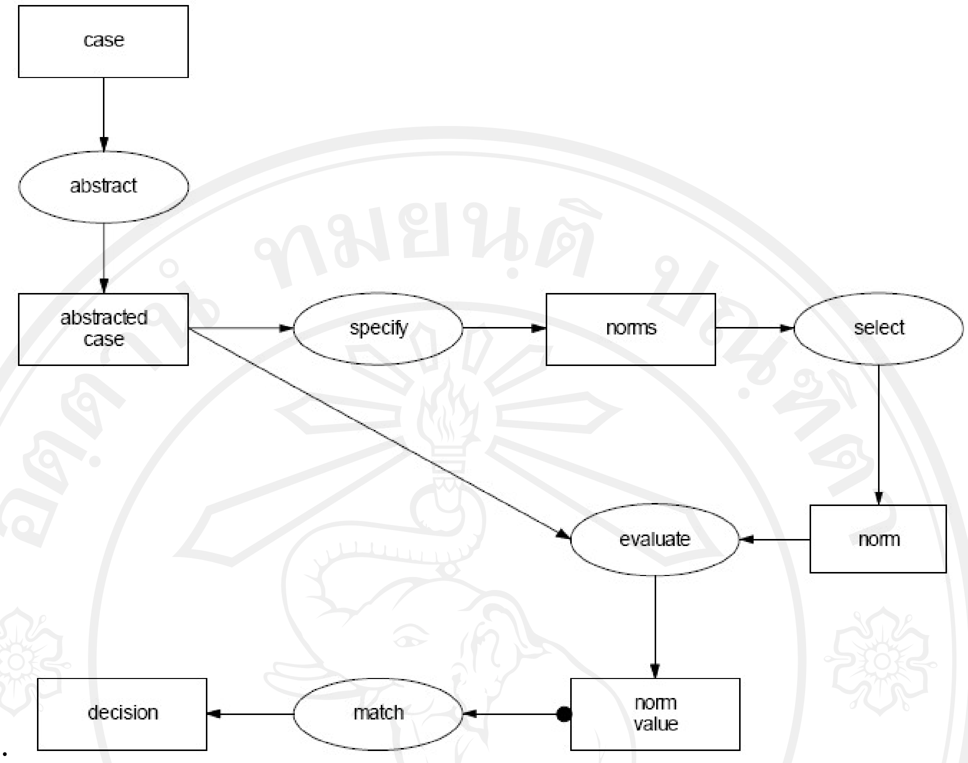

Figure 3.6 Assessment task template [38]

Table 3.8 shows the cases and corresponding norms of overhead distribution cable. Usually the utility managers expect to have their network components possessing a good characteristic (norm), thus if there is anything deviated from its norm upon the inspection, they can expect that the components may experience some form of degradation. This assessment template provides a very useful framework for asset condition assessment.

Table 3.8 Cases and norms of overhead distribution cable

Case	Norm	Deficiency
Conductor	As new	Frayed, broken, burned
Insulation	As new	Swollen, damaged, peeling, burned
Splice	As new	Leaking, swollen, imploded, sign of burning or arcing

3.4.2 Ontology101

Ontology101 is a comprehensive tutorial on the ontology development presented by Center for Biomedical Informatics Research at Stanford University [56]. Although it is proposed to supplement the usage of ontology language, Protégé 2000, but it provides a very useful framework for developing ontology and is also regarded as knowledge engineering methodology. This research also follows the procedure provided in Ontology101 for developing ontologies in distribution system asset categorization. The procedure to develop ontologies introduced by Ontology101 comprises 7 steps which include:

1) Determine the domain and scope of the ontology

By trying to answer the following basic questions: What is the domain that the ontology will cover? For what the ontology is going to be used? For what types of questions the information in the ontology should provide answer? Who will use and maintain the ontology?

2) Consider reusing existing ontologies

It is always worth considering what someone else has done and checking if such existing sources can be refined and extended for the particular domain and task. Reusing existing ontologies may be also a requirement if the intended system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies. In this research, many sources of ontologies have been studied and captured. The main sources are International Standard in Electric, MEA overhead and underground construction standard, Common Information Model (CIM) ontologies, etc. All of these standards provide useful ontologies on power distribution system.

3) Enumerate importance terms in the ontology

It is useful to write down a list of all terms which are needed either to make statements about or to explain to a user. What are the terms to be stated? What properties do those terms have? What would be an information about those terms? Initially, it is important to get a comprehensive list of terms without worrying about overlap between concepts they represent, relations among the terms, or any properties that the concepts may have, or whether the concepts are classes or slots. A better way to do this is to write down the informal description of the system that needs to be explained. For example:

- *Distribution feeder is group of many kinds of components*
- *Feeder may contain overhead or underground lines*
- *Component is a kind of asset*
- *Asset has unit price, condition rating and code*
- *Pole is a component*
- *Insulator is a component*

will provide the descriptions about distribution feeder. From above descriptions, there exist ontologies that represent concepts (e.g. asset, component, feeder, etc.), properties (e.g. unit price, condition rating, etc.) and relations (e.g. kind-of, group-of, etc.).

4) Define the classes and the class hierarchy

Concept of thing can be turned to class. The concepts obtained from informal description as shown in previous example (step 3) can turn into classes of power distribution system objects. From above example, a class of asset, component, feeder, etc., can be formulated. Furthermore, the relationships among them can form a class hierarchy. And there are several possible approaches in developing a class hierarchy; that is:

A **top-down** development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.

A **bottom-up** development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.

A **combination** development process is a combination of the top-down and bottom-up approaches: the more salient concepts would be defined first and then generalized and specialized them appropriately.

There is no particular rule to form a class hierarchy; it depends on the situation and context of modeling.

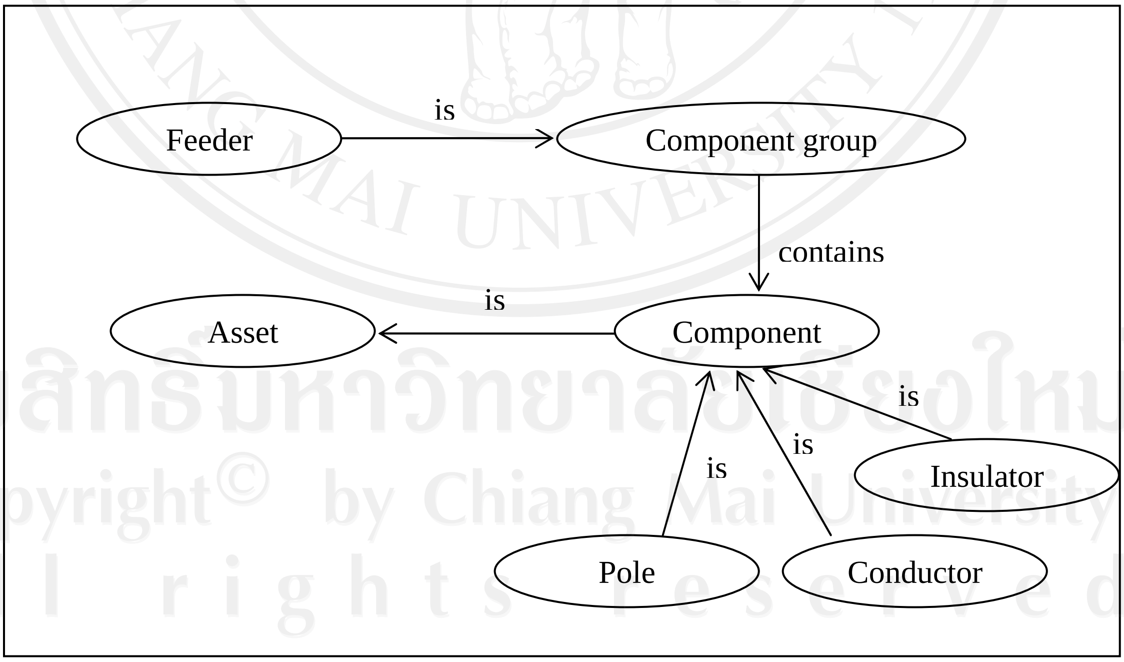

Figure 3.7 Classes and class hierarchy

Figure 3.7 shows the example of power distribution asset classes and relations among them.

5) Define the properties of classes—slots

The classes alone will not provide enough information to describe what it looks like and what it can do. Once some of the classes have been defined, it needs to describe the internal structure of concepts. This is the property or slot of the class. Slot describes the features that differentiate that class from the others. It also describes the relation it has with the others. Examples of slots of asset class may be unit price, condition rating, etc.

6) Define the facets of the slots

Slots can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take. For example, the value of a name slot (as in "the name of a Feeder") is one string. That is, name is a slot with value type String. A slot Contains (as in "a Feeder contains Pole") can have multiple values and the values are instances of the class Pole. That is, Contains is a slot with value type Instance with Pole as allowed class.

7) Create instances

The last step is creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires (1) choosing a class, (2) creating an individual instance of that class, and (3) filling in the slot values. For example, an individual instance KO-421 can be created to represent a specific type of feeder. KO-421 is an instance of the class Feeder representing all distribution feeders.

This chapter is just describing the conceptual framework of Ontology101. The actual application of power distribution system ontology development is demonstrated in chapter 4.

Throughout this research study, the knowledge engineering methodologies of CommonKADS: classification and assessment knowledge modeling as well as ontology101 are used as main methodologies to acquire and model the domain knowledge. A decision support system for power distribution asset management can be systematically developed accordingly. Since all the essential knowledge required for categorizing and making decision are already resided in-house and with the application of these techniques, the knowledge engineer can systematically capture, analyze, model, and utilize these knowledge for developing a decision support tool to assist utility manager in making an investment decision.