# CHAPTER 3

# METHODOLOGY

This chapter discusses the methodology used in the research. The concept of methodology design is to construct web-based knowledge management system by using case-based with accumulation of domain experts' semantic knowledge. The methodology is used composed of eight steps. The first step is to finding requirement of problem solving. The second step is to analysis the context of raw data to create procedure of problem solving. The third step is to capture knowledge from expert librarian. The forth step is to design the KMS framework. The fifth step is to design the algorithm are used in the KMS. The sixth step is to implement KMS in form of web-based software. The seventh step is to create domain expert involvement. And the last step is to evaluate the results. The detail of each step are describes in below section.

The chapter is organized as follows. Section 1 discusses the requirement of problem solving. Section 2 presents the context analysis of raw data to create procedure of problem solving. Section 3 presents the process to capture knowledge from expert librarian. Section 4 presents the design of framework based on KMS and CBR. Section 5 presents the algorithm of the prototype based on the following three modules to demonstrate the web application of the CBR techniques: a case retrieval module; an automated metadata generating module; and a metadata verification

module. Section 6 discusses the prototype development of the system. Section 7 presents the process to create domain expert involvement. Section 8 discusses the result evaluation.

## 3.1 Finding Requirements of Problem Solving

The objective of this step is to explore the process of metadata extraction in the library. And the selected technique is direct observed in focus group. In this step, the requirements of solving metadata extraction problem are present. The direct observe in focus group is selected to use in the research. The group of librarian expert of Naresuan University is selected to focus group. The two weeks observation in library of Naresuan University indicated that the average time to extract metadata from Thai thesis of expert librarian is around 10 -15 minutes per one thesis book. Everyday, the author writes the step of work of expert librarian to extract Thai metadata from thesis in diary. Finally, the result of this step is raw data (in form of diary) of working in metadata extraction.

## 3.2 Context Analysis of Raw Data

The objective of this step is to create the procedure of problem solving by using Thai discourse analysis techniques. In this step, the Thai discourse analysis is selected to use in transform raw data into procedure of problem solving. The results of this step which is procedure of problem solving in metadata extraction composed of 5 steps: (1) Detecting metadata keywords such as author, advisor and department

name. (2) Comparing metadata keyword with the similar words such as author can be replaced with student name. (3) Taking metadata of the keyword to store in repository. (4) Comparing new metadata with the old cases in case of expert librarian does not sure about the result. (5) Do until to get the complete metadata set.

## 3.3 Knowledge Capture from Expert Librarian

The objective of this step is to create tacit knowledge of the author in Thai metadata extraction. And the selected technique is done and rechecks with expert. This step transforms procedure of metadata extraction to tacit knowledge of the author. In this step, the author tries to manually extract metadata from Thai thesis by using the procedure from previous step. Then the results are evaluated by expert librarian of Naresuan University. The author does this routine with random 100 Thai theses. Finally, the results correctness of the author is acceptance. The result of this step is tacit knowledge to extract Thai metadata from Thai thesis of the author.

## 3.4 KMS Design



This step presents the design of KMS framework which is integrated CBR techniques. The reason why the research used CBR technique is that CBR techniques can be applied to resolve new problems by applying retrieved cases stored in the case repository and apply previous solutions to new problems. Literature review in chapter 2 shows that CBR techniques can be applied to KMS to allow knowledge sharing and reuse to achieve. In addition, web-based technology is proposed to assist in the

development of KMS. KMS cycle of Turban and Aronson (2001) is used as a basis in the framework. In the approach, knowledge is stored as cases in the knowledge database.

There are four phases in the CBR cycle: retrieve, reuse, revise and retain, and six phases in the KMS cycle. There are similarities between CBR and KMS cycle. Table 3.1 compares the phases of CBR and KMS cycle. The retrieve phase in the CBR cycle matches with the create knowledge and capture knowledge phases in the KMS cycle. The reuse and revise phases in CBR corresponds to the refine knowledge phase in KMS. Finally, the retain phase in the CBR cycle matches with the store, manage and disseminate knowledge phases in KMS.

Table 3.1 CBR cycle VS. KMS cycle.

| **CBR cycle** | **KMS cycle** |
|---|---|
| Retrieve | Create knowledge, Capture knowledge |
| Reuse | Refine knowledge |
| Revise | |
| Retain | Store knowledge, Manage knowledge, Disseminate knowledge |



We propose to use the approach discussed by Aamodt and Plaza (1994), nearest neighbor retrieval techniques is applied to allow pattern to be matched and similar cases to be compared. According to Watson (1997), pattern matching is the process of comparing two cases to one another and then determines their degree of match. In this proposed framework, pattern matching refers to the process of comparing

attributes of target case and original case to each other and determining their degree of match. Attributes of the target and original cases are associated with problem description and solution. The nearest neighbor retrieval technique is used for similarity assessment. In similarity assessment, if similar cases are found, then the software will retrieve existing cases from the knowledge repository. This way existing knowledge is retrieved and its solution can be reused. Then, the retrieve and reuse phase of the CBR cycle is implemented in this instance.

If similar case is not found, a learning algorithm will be deployed. The learning algorithm allows new knowledge to be learned based on new case behavior. For example, an element of the previous solution can be substituted or inserted with new element. This way new knowledge represented in the form of new cases with new problem and solution descriptions can be added to the knowledge data base.

However, before new knowledge is stored in the knowledge data base, the proposed result verification from expert will be deployed first. The task of verification from expert is to optimize the retrieval and accessibility of new knowledge in the knowledge data base to allow fast and efficient retrieval of knowledge, during this phase, new case is stored in the knowledge data base. Then the revise and retain phase of the CBR cycle is implemented in this instance.



The prototype KMS web application allows the above processes to be executed independently and automatically over the standardized web platform. The retrieval module is used to find the best or the closest matches case in the data base. The reuse

module allows the solution from past cases to be applied. The revise module is used to adapt a new case so that it can be retained and stored in the knowledge data base.

In CBR design, we use agents to carry out the tasks allocated in our prototype. The software agents are used in the following tasks: retrieval, reuse and revise of the CBR cycle. Figure 3.1 shows a picture of prototype process in which agents are used.

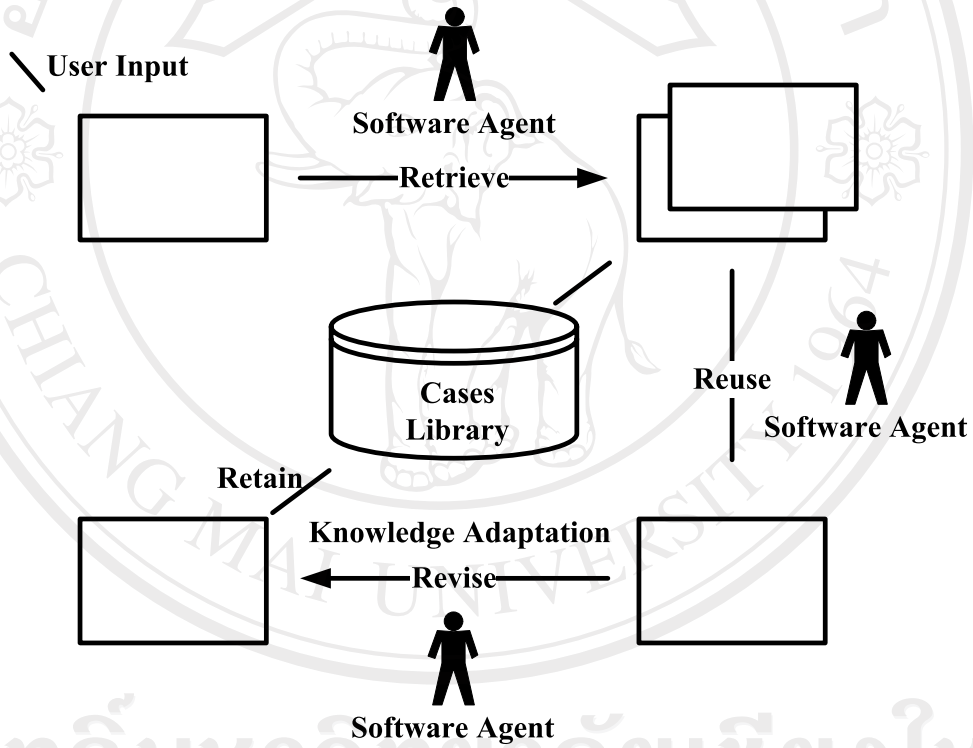


Figure 3.1 Prototype processes which agents are used.



The retrieval agent that performs similarity assessment is developed by using RDF Data Query Language (RDQL) of Jena Semantic Web Toolkit (HPL, 2002) to find the best or the closest matched case in the knowledge repository. Jena Semantic Web Toolkit is a Java Application Programming Interface (API) developed by

Hewlett-Packard Laboratory (HPL). Jena Semantic Web Toolkit includes built-in support for RDF containers, integrated RDQL, and support for storing ontology in form of model.

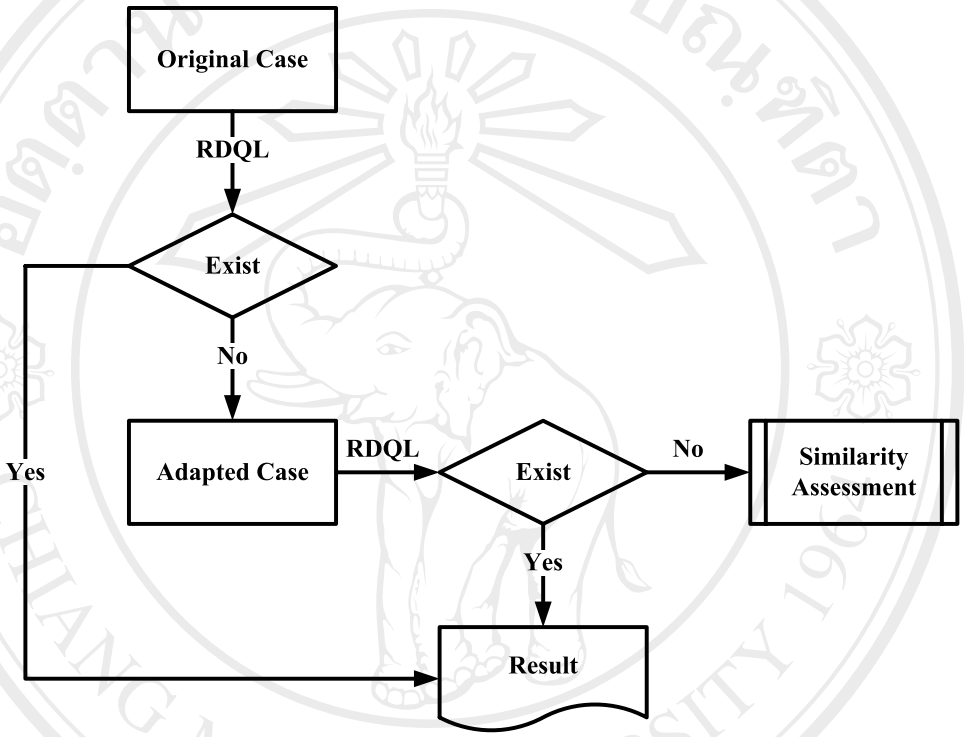


Figure 3.2 Flow chart for case retrieval.

Figure 3.2 shows the flowchart for the retrieval phase of the CBR. The user query is performed by using RDF query. The system considers each user query as a new case (problem case). In general, there are two types of cases stored in the case repository composed of the original case and the adapted case. The original cases are cases that were given by the domain expert before the system is developed. In our study, the domain expert has provided one hundred original cases. On the other hand, the adapted cases are cases that have been adapted by the system as a result of the

revise and retain phase of the CBR cycle. Note that the original cases have precedence over the adapted cases in the retrieval process. Figure 3.2 shows that the system will first attempt to find an existing original case that matches with the target case. If a matched original case is found, then the system will return the outcome based on the solution component of the original case. Otherwise, the system will attempt to find a matched adapted case. If one is found, then the solution of the matched adapted case will be returned. If none of the matched original and adapted cases are found, the system will process similarity assessment, and the case that is found to be the nearest to the target case will be retrieved from the knowledge repository.

The similarity assessment is performed using the Nearest Neighbor technique. It processes retrieval of cases by comparing a list of weighted attributes in the target case to source cases in the knowledge repository. Weighting is assigned to the attributes in the form of relative importance. In our prototype, the domain expert determines the weighting of each attribute. In the prototype, the university names and the faculty name are considered twice as important compare to other attributes and are given a weighting of 2. Note that the weight is assigned an integer value to simplify the calculation. The formula of Nearest Neighbor Retrieval calculates the distance of the target case from the original case.



After the case is retrieved, the reuse agent will apply the solution from the retrieved case to solve the target case by recommending whether the applicant is eligible for the course she/he intends to apply when the matched or nearest matched

case is found. The reuse phase of the CBR cycle allows the solution to be modified and the target case to be adapted so that is can be retained and stored in the knowledge repository as new case. However before case can be retained, the case must be verified and revised using knowledge adaptation technique.

For the revise phase, we use derivation adaptation technique in our prototype. An agent is developed in our prototype to perform this task. This is the function of the learning agent process which is related to the phase of Refine Knowledge of KMS cycle. Derivational adaptation is a technique to reuse the rules of formals that generated the original solution to produce a new solution to the current problem (Watson 1997).

In the retain phase, indexing is performed when cases are stored in the knowledge repository. Indexing allows cases to be retrieved more efficiently. This phase is related to a software agent that performs Store Knowledge, Manage Knowledge and Disseminate Knowledge phases in the terms of KMS cycle which.

The result of this step is the architecture of the system. In practical, the prototype system composed of three modules: case retrieval module for comparing problem case and stored case, metadata generating module for applying solution from past case to automatically extracting metadata from electronic Thai documents, and metadata verification module for identifying and correcting the errors in extracted metadata. The architecture of the prototype is used in this research as shown in figure 3.3.

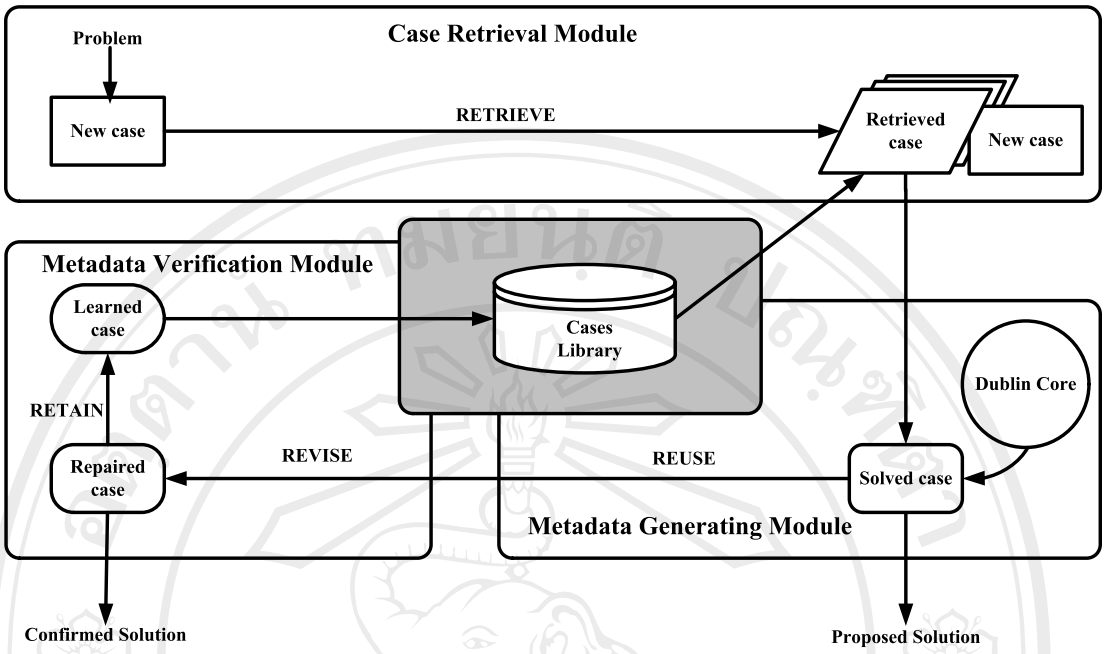


Figure 3.3 Architecture of the prototype system.

The above framework will be implemented as a metadata extraction system which is a web-based application. The prototype will apply metadata over the web based platform. An advantage of using metadata is to define the concepts relevant to domain of knowledge. One of the most common goals of metadata development is to share common understanding of the structure information and to enable sharing and reuse of domain knowledge among people and software system.



## 3.5 Algorithm Design

This section presents the algorithm of the prototype. The design concept of the algorithm is to emulate CBR cycle process. Normally, CBR cycle composed of four phases, but the algorithm design of the research composed of three modules. In the

algorithms design, the revise and retrieved phase can be considered as the identical module. The prototype is based on three modules to demonstrate the CBR techniques in prototype development composed of case retrieval module, automated metadata generating module and metadata verification module.

### 3.5.1 Case Retrieval Module

A case classification module is used for a comparing problem case and a stored case using Nearest Neighbor Retrieval (NRR) technique. NRR is a technique to measure how similar the target case comparing to a source case. It processes retrieval of cases by comparison of a collection of weighted attributes in the target case to source cases in the CBR library. If there is no matched case in the CBR library, CBR system will return the nearest matched source case. The return of the nearest case match can be represented by the following Equation.

$$\text{Similarity}(T,S) = \sum_{i=1}^{n} f(Ti,Si) * Wi$$

Where T is the target case, S is the source case, n is the number of attributes in each case, i is an individual attribute from 1 to n, f is a similarity function for attribute I in cases T and S and W is the importance weighting of attribute I

The equation of the NNR represents the sum of similarity of the target case to the source case for all attributes multiplied by the importance weighting of individual

attributes. The CBR system therefore retrieves a meaningful case that may provide a detailed solved problem description to a new problem.

### 3.5.2 Metadata Generating Module

A metadata creation module is responsible for automatically extracting the metadata from Thai electronic documents. In this framework, Thai keywords that matched with Dublin Core categories will be extracted into database. To work with heterogeneous documents, the template-based approach, as a set of rules, is proposed to classify documents into a group. For example the header of student thesis abstract, as shown in figure 3.4, is roughly detected nine keywords which are thesis title, student's name, advisor's name, advisor's degree, degree name, major name, department name, faculty name and graduate year.

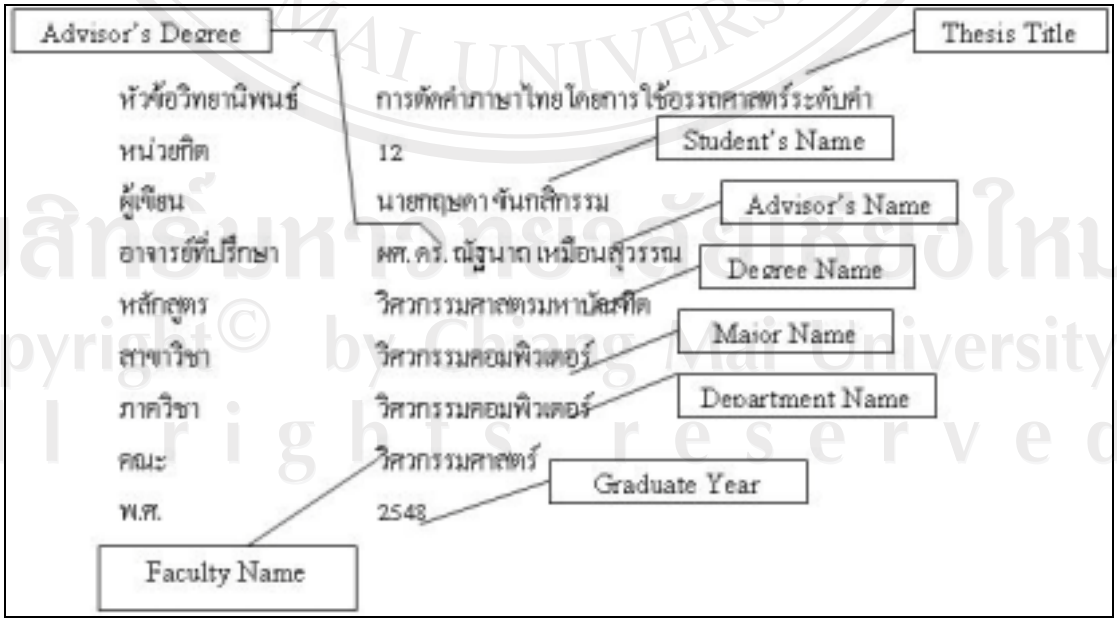


Figure 3.4 Example of Thai theses abstract and its structure.

After analyzing the header of thesis abstract by using template-based approach, the system can easily identify the boundary of each part by using delimiter (e.g. ‘:’) and keyword markers (e.g. |หัวข้อ|, |ผู้เขียน|, |อาจารย์ที่ปรึกษา|, |สาขาวิชา|, |ภาควิชา|, |พ.ศ.|).

Thus, the system can create an analyzed structure of thesis abstract by using those boundary markers as a part separation point. The example of created analyzed structure is represented in figure 3.5. After analyzing the abstract header, the metadata can be directly extracted from analyzed results.

| | |
|---|---|
| \|หัวข้อ\|วิทยานิพนธ์ | :การตัดคำภาษาไทยโดยการใช้อรรถศาสตร์ระดับคำ: |
| หน่วยกิต | 12 |
| \|ผู้เขียน\| | :นายกฤษดา ขันกสิกรรม: |
| \|อาจารย์ที่ปรึกษา\| | ผศ. :ดร.: :ณัฐนาถ เหมือนสุวรรณ: |
| \|หลักสูตร\| | :วิศวกรรมศาสตรมหาบัณฑิต: |
| \|สาขา\|วิชา | :วิศวกรรมคอมพิวเตอร์: |
| \|ภาควิชา\| | :วิศวกรรมคอมพิวเตอร์: |
| \|คณะ\| | :วิศวกรรมศาสตร์: |
| \|พ.ศ.\| | :2548: |

Figure 3.5 Example of the thesis abstract header analyzed structure.

After extracting the metadata, the metadata verification module will help users identifying and correcting the error in extracted metadata in order to obtain a high quality metadata.

### 3.5.3 Metadata Verification Module

The extracted metadata may contain errors both from the metadata creation module and original documents. To gain a high precision metadata, it is necessary to identify and correct the errors before using the metadata (Kawtrakul & Yingsaeree, 2005). The proposed framework is an integrated mechanism in order to help users to correct the errors. Regarding to errors in metadata creation module, the system may not be able to extract some documents due to incompleteness of source case or defect in the documents. In this process, the system will display error messages from metadata creating module to guide users for correcting the errors. The users make a decision to response with the errors. There are many sophisticated methods that can be employed in order to help the users detecting and correcting the errors. The good choice is to use a spelling correction technique (Theeramunkong & Usanavasin, 2001) to detect errors and suggest the correction.

## 3.6 Software Implementation

In this section, the prototype development is presented. The prototype is an automated Thai metadata extraction system that allows domain expert and expert librarian to revise the proposed result from the system.



The prototype is implemented by PHP (Hypertext Processor) language and MySQL is selected to repository of the prototype. In the prototype, knowledge is stored in repository as a solution of previous problem. Figure 3.6 shows the

knowledge that stored in the knowledge repository. The metadata are stored as a problem description in the case (see figure 3.7). The knowledge repository consists of two types of result, the proposed results and the confirmed results. The proposed results are provided by the prototype system (an automated metadata extraction module) and the confirmed results which the result are retained in the knowledge repository. These cases, previous solution of problem and results are stored in MySQL database management system.

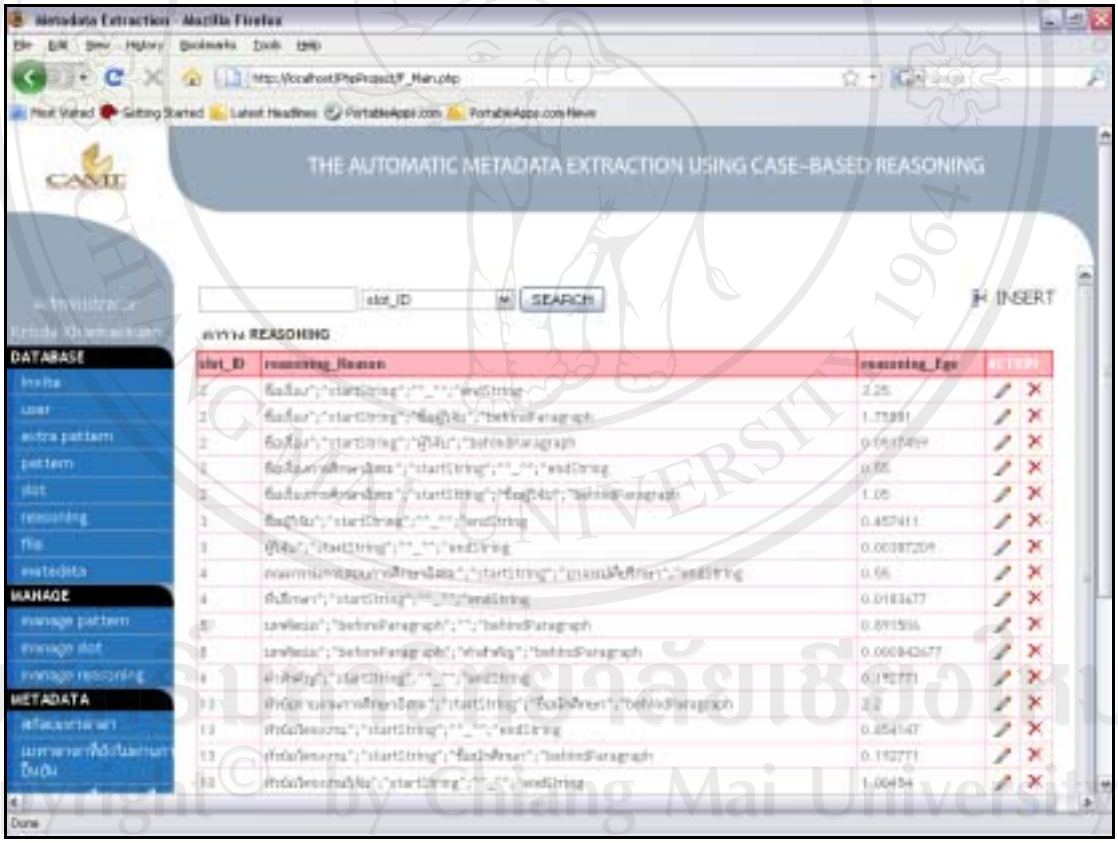


Figure 3.6 A reasoning screen.

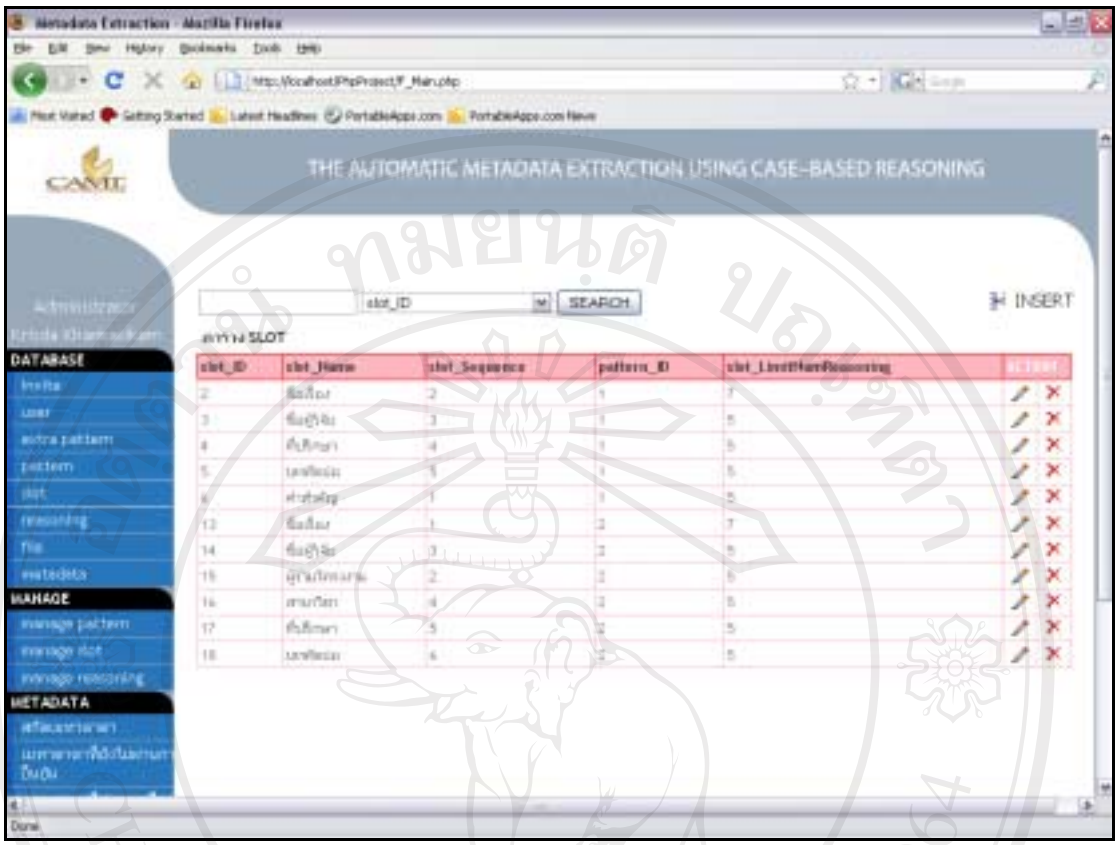


Figure 3.7 An attribute of metadata screen.

Each user input (problem case) is stored in Microsoft Office Word format, then the system is transformed the MSOW file to text file using the WordToText application. The input needs to be transformed to text file format because Hypertext Preprocessor (PHP) language can process only text file. Figure 3.8 shows a sample of user input screen.

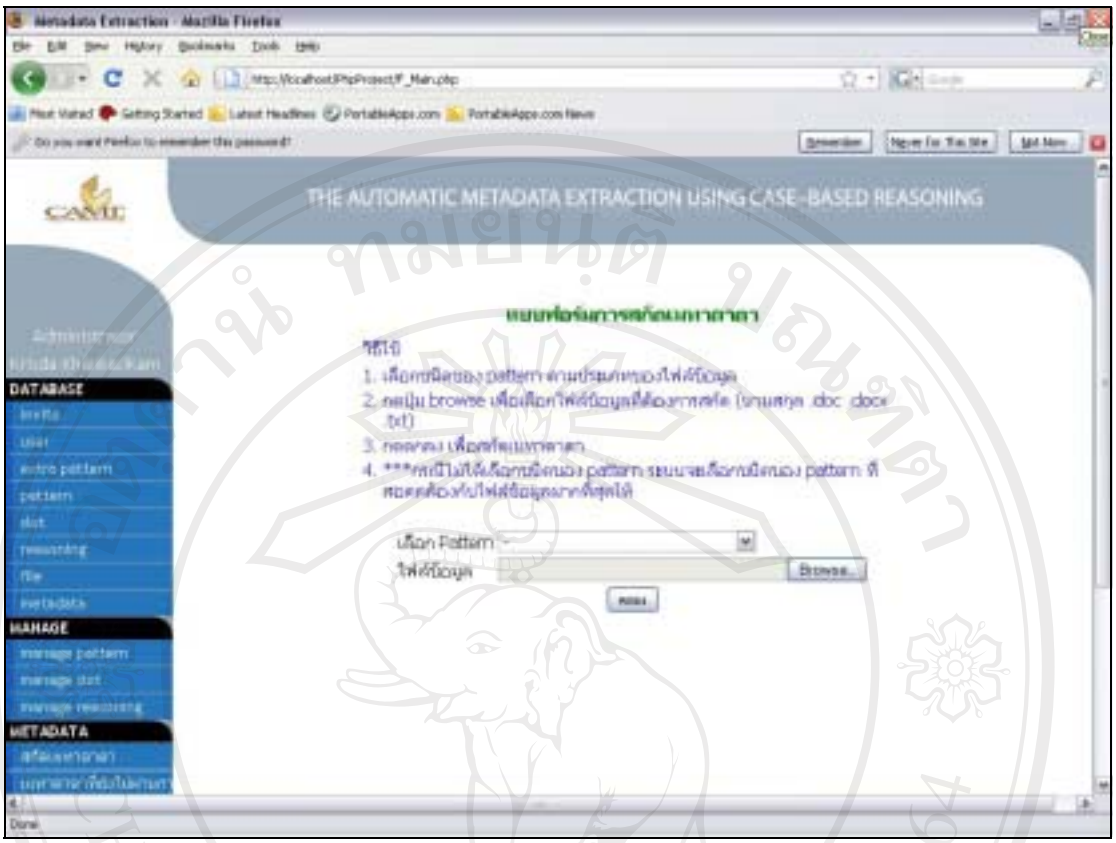


Figure 3.8 A user input screen.

The attributed that are relevant to the problem description (e.g. author, advisor, abstract and year) will be extracted from the user input screen and will be stored in the database system so it can be retrieved when the user required this information. Figure 3.9 shows a sample of metadata retrieval screen.

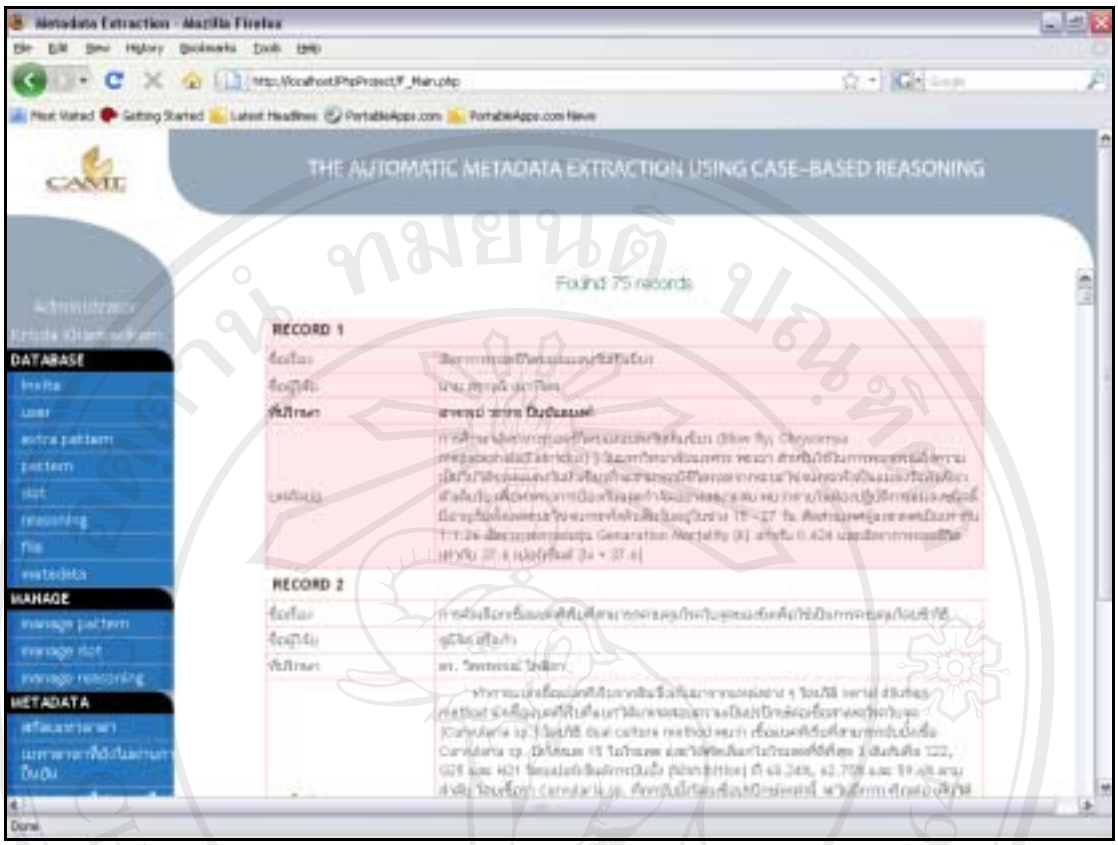


Figure 3.9 A user metadata retrieval screen.

Each of the domain expert and expert librarian input will become a repaired case in the system. Figure 3.10 is represented the proposed resulted from the prototype system that waiting for conformation from domain expert or expert librarian. This repaired case may become a target case and can be stored in knowledge database after they have been edited in the revise phase of the CBR cycle.

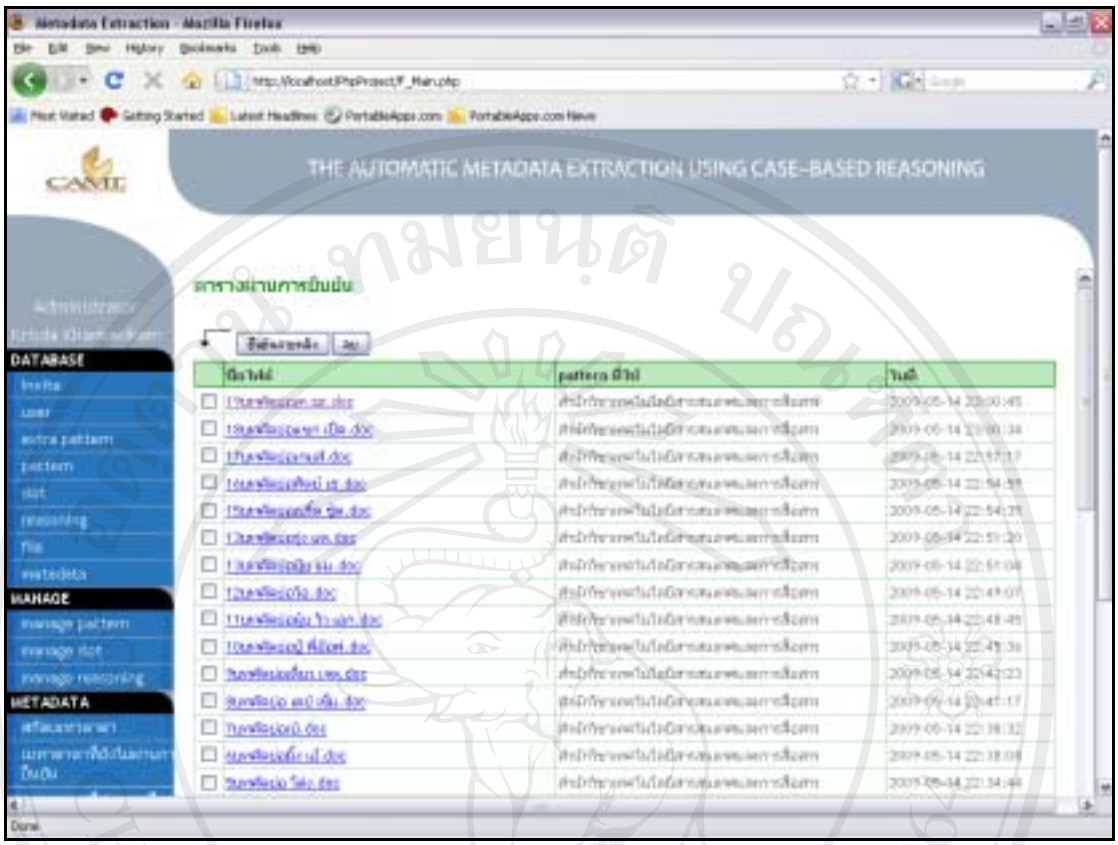


Figure 3.10 A proposed result screen.

## 3.7 Domain Expert Involvement

This step presents the collaboration with domain experts. There are collaboration from domain experts in agriculture of Faculty of Agriculture, Naresuan University which are Associated Professor Dr. Chada Narongrhit and Associated Professor Dr. Duangporn Premjit. The domain experts revise the results of proposed metadata that automated produced by the computer system. If there is some mistake in the proposed results the domain experts will edit these metadata and then the system can learn from the domain expert and retain new knowledge in knowledge

repository. If there is not mistake in proposed results the domain expert will confirm the proposed solution and then the proposed results became the confirmed results.

## 3.8 Result Analysis

The evaluation of the system can be divided into two levels composed of qualitative and quantitative approach.

### 3.8.1 Evaluation of System Using Qualitative Approach

In qualitative approach, there is collaboration from three groups of users composed of domain experts, expert librarians and general users to evaluate the satisfaction of using the prototype. Domain expert group composed of two experts in agriculture field of Naresuan University. Expert librarian group composed of four expert librarians from Naresuan University and Nakhon Sawan Rajabhat University. And general user group composed of 500 students of Nakhon Sawan Rajabhat University. The questionnaires are used for evaluation the following scenarios: (1) the correctness of the prototype system result; (2) the easy usage of the prototype system; (3) the saving time of the prototype using; (4) the over all of satisfaction.



### 3.8.2 Evaluation of System Using Quantitative Approach

In quantitative approach, the typical approach for evaluating a metadata extraction system is to create an ideal metadata by expert for comparing with the

results. Unlike English, standard data set in Thai are not yet available for evaluating metadata extraction system. However, in order to observe characteristics of our prototype, we collected Thai theses which content related to sufficient economy and Thai folk wisdom, including 2,000 theses from Naresuan University to make data sets. In data analysis of this study, there is collaboration from experts in metadata extraction of Naresuan University library to check the correctness of system results. The confirmation or editing from the expert will gradually affect to the increasing of system intelligence.

Our experiments with prototype of Thai metadata extraction by using case-based reasoning are performed on the Thai theses which content related to sufficient economy and Thai folk wisdom. These theses are written in originally Thai language without using words originate from English language such as computer-คอมพิวเตอร์ or words standardize by The Royal Institute (TRI), an institution concerns academic matters as the compilation and publication of dictionaries, encyclopedias, terminologies, taxonomies, because this system and its techniques can process data using original Thai words only.

We evaluated results of metadata extraction prototype system by using three widely used methods (Baeza-Yates and Ribeiro-Neto 2002) composed of Precision, Recall and F-measure indices. Let R*a* is the number of correctness extracted metadata, A is the number of ideal extracted metadata and R is the number of extracted metadata in actual answer. Precision of the algorithm can be calculated as

the fraction between the numbers of correctness extracted metadata and the number of ideal extracted metadata. In this research work, the Precision index is defined as

$$\text{Precision} = \frac{\text{R}a}{\text{A}}$$

Then, the Recall index is the fraction between the numbers of correctness extracted metadata and the number of extracted metadata in actual answer; that is,

$$\text{Recall} = \frac{\text{R}a}{\text{R}}$$

Finally, the average value of the Precision and Recall indices called "F-Measure index" can be calculated as follows:

$$F_{\beta} = \frac{(1+\beta^2) * \text{Precision} * \text{Recall}}{\beta^2 * (\text{Precision} + \text{Recall})}$$

β is a factor which is used to adjust weight between Precision and Recall indices. Two other commonly used F-measures are the $F_2$ measure which weights Recall twice as much as Precision and the $F_{0.5}$ measure which weights Precision twice as much as Recall. In this research, β is defined as 1 because Precision and Recall are evenly weighted. This $F_1$ measure is also known as the Harmonic mean specified by

$$\text{F-measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

According to the definition of the previous three measures, these measure indices are direct performance measures how well metadata extraction does. In general, the Precision index is used to indicate direct correctness of metadata extraction. For both Recall and F-measure indices, they do not directly indicate metadata extraction performance. However, they are useful for indirect correctness measure of metadata extraction. Either Recall index or F-measure index is one of the most important performance measures in information retrieval system. Actually, the F-measure index represents the compromise between the Precision and Recall indices.