Software Requirements Retrieval Using Use Case Terms and Structure Similarity Computation

Akadej Udomchaiporn[1], Nakornthip Prompoon[2], and Pizzanu Kanongchaiyos[2]
[1]*Department of Mathematics and Computer Science, Faculty of Science*
King Mongkut's Institute of Technology Ladkrabang
[2]*Center of Excellent in Software Engineering, Department of Computer Engineering*
Faculty of Engineering, Chulalongkorn University
kuakadej@kmitl.ac.th *nakornthip.s@chula.ac.th* *pizzanu@cp.eng.chula.ac.th*

Abstract

For a large scale of software development, there is a tremendous number of software requirements documents in a collection which may be produced for different domains by different developer teams. They may be later reused to reduce cost and time for the next development. Thus, there is a need to retrieve ones that meet user's need efficiently. This paper presents an approach for software requirements specification retrieval in a form of use case description using use case structure and similarity computation between terms of use case query and use cases in the collection. The contribution of the paper has five main points; 1) the approach for retrieving use case description is proposed, 2) the developed tool supporting the approach is presented, 3) the experiment is designed to measure effectiveness of the approach, 4) the results of the experiment are shown to compare effectiveness of the approach to that of a general approach, and finally 5) the recommended use case description query template is proposed.

1. Introduction

Nowadays, there is a high competition among software developers in software development industry. Therefore, software developers have to find ways to reduce cost of software development process as much as they can in order to reduce software production cost to gain competition advantage in the software market. Software reuse [1-3] is one of efficient approaches to support this idea. There are many software components which can be reused, and each of them may be a consequent product from early phases of software development process. Therefore, reusing software components is more efficient when they are reused in early phases of software development process such as requirements analysis phase because their consequent products in the following phases can be identified and reused too. This finally leads to reduce time and cost of software production and to increase software quality.

In requirements analysis and design phase, UML (Unified Modeling Language) [4, 5] is frequently selected to represent user requirements and software characteristics from many points of view. Moreover, it helps developers to plan activities in the next phases efficiently. UML is composed of many kinds of diagrams, but the important one which is used for capturing user requirements and for indicating functional requirements of software systems is a use case diagram. Moreover, it can also be used as an agreement between users and developers. Each use case has a use case description which collects details of it. A use case description counts as a software component in requirements analysis phase which can be efficiently reused [6]. Due to the fact that contents of a use case description are collected in a natural language, information storage and retrieval theories can be applied. Therefore, some theories such as automatic indexing, term weighting system, similarity computation, and retrieval evaluation are applied to our approach.

The rest of this paper is organized as follows; related work and background knowledge are introduced in section 2 and 3. The approach for retrieving use case is described in details in section 4. The experiment and its results are shown in section 5. Finally, summary and future work are summarized in section 6.

2. Related Work

In research conducted on this issue at the moment [7-11], there are many related works about use case retrieval, but they still have some disadvantages such as complex process, slow response of retrieval system, and manual process. Woo [11] proposed an automated technique that provides for the assisted reuse of use cases. They developed a tool named "ScenAsst" for collecting and retrieving use case conveniently. ScenAsst transforms contents of use cases into graph, clusters them, and then stores them into the use case collection. In retrieval process, user's query is also transformed into graph. The similarity between a graph from user's query and each graph collected in the collection is compared by

XIII ASIA PACIFIC SOFTWARE ENGINEERING CONFERENCE (APSEC'06)
0-7695-2685-3/06 $20.00 © 2006 IEEE

"SUBDUE" algorithm. This technique causes slow response in retrieval process because of its complexity. Blok [7] presented an approach for reusing UML specifications focusing on use cases. In their storage process, a set of indices from use case descriptions is identified and clustered by the experts. In retrieval process, user's query is transformed into a set of indices. A set of indices from user's query and from each use case in the collection are computed to be a similarity score. This approach has the experts manually generate indices of a use case for both storage and retrieval process, so the retrieved results are rather precise. On the other hand, the retrieval process depends on experience of the experts, so the retrieval results may be bias. Furthermore, the users cannot develop a tool supporting this process automatically. For other research, Saeki [8, 9] proposed a pattern for reusing requirements specification. Requirements specification received from his pattern can be collected in a form of use case descriptions in the use case description template that he also presented. However, this research can reuse only the pattern for getting requirements specification, but it cannot reuse contents of that requirements specification.

Meanwhile Porter's algorithm [12] used for transforming words into their grammatical root and Frake's stop list [13] used for eliminating non-significant words are work products on information storage and retrieval which were utilized in this paper. These work products help our approach increase effectiveness of our retrieval system.

3. Background

3.1 Use case description

Use case description is a document describing details of use cases in use case diagram and it is written in a natural language. Use case description fundamental components [4, 14, 15] which cover important contents and which are generally used consist of nine components. They are

1) ***Use Case Name*:** Use case name is a name of use cases. Every use case must have a name that distinguishes it from other use cases.

2) ***Objective*:** An objective is an element describing about an objective of use cases.

3) ***Actor*:** An actor is someone or something outside the system that interacts with the system.

4) ***Relationship*:** Relationship is a semantic connection between model elements. In a use case description, relationships are composed of association, include, extend, and generalization. Relationship components contain use case name of the use case relating with itself.

5) ***Precondition*:** Precondition is a constraint that must be true when a use case is invoked.

6) ***Postcondition*:** Postcondition is a constraint that must be true when a use case has ended.

7) ***Normal flow of events*:** Normal flow of events is the element of a use case that describes its most common implementation. The basic flow is written assuming that no errors or alternatives exist. Also called basic path or normal path.

8) ***Subflow*:** In some cases, normal flow of events can be decomposed into a set of subflows to keep the normal flow of events as simple as possible.

9) ***Alternative or exceptional flow of events*:** Alternative or exceptional flow of events is the element of a use case that describes its alternative implementation. It is also used to describe error conditions, since errors can be considered a kind of alternative. It is also called alternative path.

This use case description template is depicted as use case query by use case structure screen in figure 1.

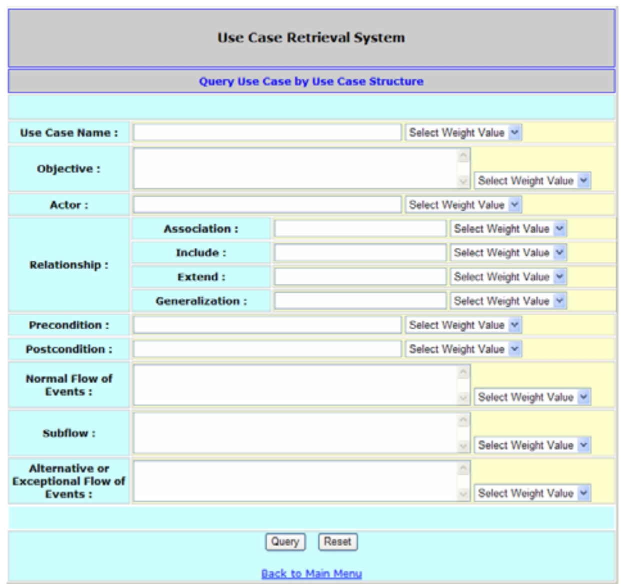


Figure 1. Use case query by use case structure

3.2 Information storage and retrieval

Salton [16] mentioned that "An information retrieval system is an information system, that is, a system used to store items of information that need to be processed, searched, retrieved, and disseminated to various user populations." In this paper, some theories of information storage and retrieval system [16, 17] such as automatic indexing, term weighting system, similarity computation, and retrieval evaluation are applied. They are briefly introduced as follows.

1) Automatic indexing: The indexing task consists of assigning to each stored item terms, or concepts, capable of representing document contents, so automatic indexing is an automatic process to determine that which terms in the document collection should be used as index terms.

2) Term weighting system: Term weighting system is the process of assigning each term a weight, or value, reflecting its presumed importance for purposes of content identification. Currently, there are many techniques for term weighting system but a simple technique used in this paper is inverse document frequency (IDF) weighting system [16].

3) Vector similarity computation: Salton [16] mentioned about vector similarity function that "In information retrieval, the objects might be documents and the properties could be the index terms, and properties could be the document identifiers to which the terms are assigned. The similarity between two objects is normally computed as a function of the number of properties that are assigned to both objects." The example of vector similarity computation widely used in literature to measure vector similarities and used in this paper is Dice's coefficients [16].

4) Retrieval evaluation: The purpose of evaluation is to measure effectiveness and efficiency of an information storage and retrieval system. For evaluation process, there are three well-known metrics; recall, precision, and harmonic mean [16, 17]. They are introduced as follows.

- **Recall** is defined as the proportion of retrieved and relevant documents to all relevant documents in the collection.
- **Precision** is the proportion of retrieved and relevant documents to all retrieved documents.
- **Harmonic mean** is a single measure which combines recall and precision.

4. Use Case Storage and Retrieval Process

Use case storage and retrieval process in our approach consists of three main processes being 1) storage, 2) retrieval, and 3) evaluation process. A tool has developed to support retrieving use cases automatically. The activities are divided to 3 steps. They are briefly introduced as follows.

Step 1: Use case collection and index creation

Use cases developed from example domains are collected and transformed into a set of indices and their weighted value.

Step 2: Query generation and retrieval process

A user generates use case queries and their weighted value in the predefined use case format shown in figure 1, and those queries are also transformed into a set of indices. After that, a set of use cases in the use case collection is retrieved according to their similarity with the user's use case query, and it is presented to the user.

Step 3 : Retrieval evaluation

The retrieved use cases are evaluated which one is relevant or irrelevant to the user's use case query in order to evaluate the approach by computing recall, precision, and harmonic mean.

4.1 Use case collection and index creation

Use case collection and index creation consists of two main steps; they are

1) Automatic indexing

After collecting use cases, their contents written in natural English language are parsed and transformed into a set of indices using automatic indexing process [16, 17]. Its steps are described as follows.

- **Parsing contents from use cases** is to separate each word from the use case contents.
- **Eliminating words from the stop list** is removal of stopwords from a stop list because these words are poor discriminators and cannot be possibly used to identify use case content. In English, about 425 common words are involved, and they are included in the stop list [13].
- **Stemming words into its grammatical root** is to reduce the original words to word stem for reducing a variety of different forms. One of the favorite models for stemming words is Porter's algorithm [12].
- **Eliminating high frequency words** is to eliminate high frequency function words which their frequency is more than a predefined threshold because they are considered as poor discriminators.

2) Term weighting system

After getting a set of indices from automatic indexing process, each index is weighted by using inverse document frequency (IDF) technique. Finally, all indices and their weighted value are stored into the collection.

4.2 Query generation and retrieval process

Firstly, a user has to generate a simple use case as a query in a predefined form shown in figure 1. The user's use case query is transformed into a set of indices by automatic indexing process which is the same as those in the index creation step. A user also defines a weighted value of each element of use case query. These weighted values which are integers in 1-5 intervals are used as a factor in similarity computation. Similarity scores between each element of user's use case query and each element of each use case collected in the collection are computed by equation (1).

$$Similarity(E_{mi}, E_{mj}) = \frac{2\left[\sum_{k=1}^{t}(Term_{mik} \bullet W_{ik}) \bullet Term_{mjk}\right]}{\sum_{k=1}^{t} Term_{mik} + \sum_{k=1}^{t} Term_{mjk}} \quad (1)$$

Where:

E_{mi} = element *m* of use case *i*.

E_{mj} = element *m* of use case *j*.

$Term_{mik}$ = 1 when term *k* appears in element *m* of use case *i*,

= 0 when term *k* does not appear in element *m* of use case *i*.

$Term_{mjk}$ = 1 when term *k* appears in element *m* of use case *j*.

= 0 when term *k* does not appear in element *m* of use case *i*.

W_{ik} = IDF weighted value of term *k* in use case *i*.

Similarity scores between user's use case query and each use case in the collection are computed by equation (2).

$$Similarity(UC_i, Query_j) = \frac{\sum_{m=1}^{e} \left[Similarity(E_{mi}, E_{mj}) \bullet WE_m \right]}{TotalWeight} \qquad (2)$$

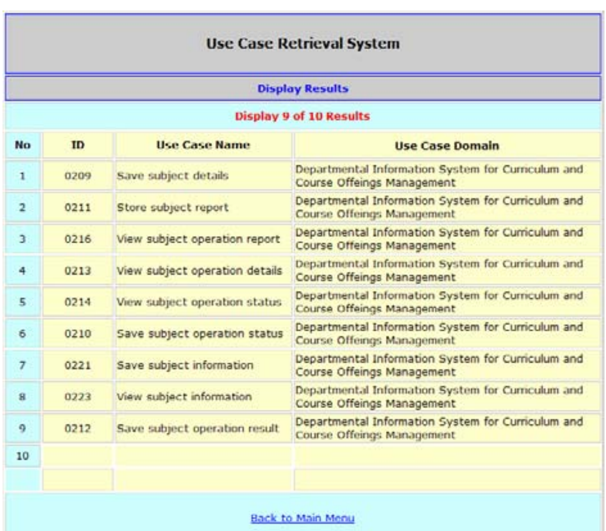
Where:

$Similarity(UC_i, Query_j)$ = similarity score between user's use case query and each use case in the collection.

$Similarity(E_{mi}, E_{mj})$ = similarity score between each element of use case *i* and query use case *j* computed by equation (1).

WE_m = weighted value of element *m* defined by the user.

$TotalWeight$ = summary value of weighted values (WE_m) of all components.


Use Case Retrieval System

Display Results

Display 9 of 10 Results

No	ID	Use Case Name	Use Case Domain
1	0209	Save subject details	Departmental Information System for Curriculum and Course Offeings Management
2	0211	Store subject report	Departmental Information System for Curriculum and Course Offeings Management
3	0216	View subject operation report	Departmental Information System for Curriculum and Course Offeings Management
4	0213	View subject operation details	Departmental Information System for Curriculum and Course Offeings Management
5	0214	View subject operation status	Departmental Information System for Curriculum and Course Offeings Management
6	0210	Save subject operation status	Departmental Information System for Curriculum and Course Offeings Management
7	0221	Save subject information	Departmental Information System for Curriculum and Course Offeings Management
8	0223	View subject information	Departmental Information System for Curriculum and Course Offeings Management
9	0212	Save subject operation result	Departmental Information System for Curriculum and Course Offeings Management
10			

Back to Main Menu

Figure 2. Use case retrieval result form

The results which are use cases in the collection and have topmost *T* similarity scores (*T* is the predefined threshold number) computed from equation (2) are displayed to the user. An example of use case retrieval result form is depicted in figure 2.

4.3 Retrieval evaluation

In retrieval evaluation, the user has to identify all use cases in the collection which are relevant to his/her use case query in order to compute a value of three selected metrics which are recall, precision, and harmonic mean. The metrics are used to measure effectiveness of the retrieval system, and they were introduced in section 3.2.

5. The Experiment

The main objective of the experiment is to test our research assumption that whether use case retrieval by use case structure is more effective than use case retrieval by only use case keywords. Therefore, the experiment is provided in order to compare effectiveness of our approach, use case retrieval by use case structure, and effectiveness of a general approach, use case retrieval by only use case keywords. Three selected metrics which are recall, precision, and harmonic mean are used in effectiveness measurement in the experiment.

5.1 Controlled variables

This experiment is designed for eliminating bias, so it has many controlled factors. These factors are separated to four variables, use case collection, users, search process, and queries. They are described as follows.

1) Use case collection: 315 use cases from 16 use case domains are selected from well-designed systems, and they were written in English language.

2) Users: This experiment has 10 users to test the use case retrieval system. For all users, everyone has studied in the Master's degree program in software engineering field of study. Therefore, they have had experience and capability about use case modeling, and their knowledge about English language is quite fair because English score of all testers is about 500 for TOEFL [18] equivalent score.

3) Search process: Five sample subjects are set for users to test the use case retrieval system. The users have to search use cases from 16 use case domains by generating search statements which are keywords relevant to five given sample subjects; they are

A) Teaching-studying system
B) Product trading system
C) Customer information management
D) Financial calculation
E) Report generation

Answer sets of use cases for each subject are identified in advance and then collected in the database in order to compute recall, precision, and harmonic mean.

The objective of five different sample subjects is to compare effectiveness of both use case retrieval systems, use case retrieval by use case structure and by use case keywords in two points of view. There are view of functional requirements and a system domain. The reason of giving sample subjects in two points of view is because we want to observe broad and narrow subjects whether they have an effect on effectiveness of use case retrieval in the approach. Subject A and B are in a view of system domain because they are not identified functional requirements, but they just indicate their wanted system domains. Meanwhile, subject C and D are in a view of both functional requirements and a system domain because they are identified both functional requirements and their system domain. Therefore, scope of subject C and D are narrower than subject A and B. The last subject, E is in a view of only functional requirements because it is identified functional requirements or main function of the system, but the system domain is not identified. The reason of giving five sample subjects is for variety of users' query which may eliminate their preference bias.

4) Queries: The users have to generate search statements for queries both by use case keywords and by use case structure for all five subjects, A, B, C, D, and E. For each subject, the user can query five times. Therefore, each user has to generate 50 set of queries, 25 sets for query by use case keywords, and another 25 sets for query by use case structure. Finally, the experiment get 500 sets of queries for all users, 250 sets for query by use case keywords, and another 250 sets for query by use case structure. In other words, the experiment get 100 sets of queries for each subject, 50 sets from query by use case keywords, and another 50 sets from query by use case structure.

These are four controlled variables. A selected user can query use cases following the controlled variables mentioned above. For each subject, all 50 sets of queries from query by use case keywords and all 50 sets of queries from query by use case structure are used to query use case by the developed tool. The retrieved results of those queries are used to calculate recall, precision, and harmonic mean for comparing effectiveness of both use case retrieval systems in the experiment.

5.2 Experimental threshold

There are two main thresholds in the experiment. They are introduced as follows.

1) Similarity threshold: the similarity threshold for the experiment is defined as 0.00039. This threshold is computed by equation (3).

$$mean - \alpha \qquad (3)$$

Where:

$mean$ = average value of similarity score for all queries in the experiment (0.00077).

α = standard deviation value of all queries in the experiment (0.00038).

The reason of $mean - \alpha$ is because size of use case collection used in the experiment is small. The use case collection is composed of 315 use cases from 16 use case domains. Therefore, some use cases may fall out if similarity threshold was defined higher such as $mean$ or $mean + \alpha$. However, if the use case collection is larger, the similarity threshold can be adjusted higher.

2) High frequency threshold: In the experiment, high frequency terms are not eliminated because size of use case collection used in the experiment is small. The number of all terms in the collection is 651 terms. Thus, some significant terms may be lost if high frequency threshold was defined higher. However, if the use case collection is larger, the high frequency threshold could be adjusted higher.

5.3 Experimental results

The summary results of the experiment are shown in table 1. Experimental results are summarized and compared between use case query by use case structure and by use case keywords. For use case query by keywords, two characteristics of use case query by use case keywords are experimented; they are

(1) Use case query by keywords generated by users.

(2) Use case query by keywords generated from all keywords appeared in each component of use case query by use case structure.

Use case query by use case structure screen is depicted in figure 1, and use case query by use case keywords (1) screen is depicted in figure 3. Table 1 shows average recall, precision, and harmonic mean of query by keywords (1), by keywords (2), and by use case structure (3). The abbreviations in table 1 and table 4 are introduced as follows

- *R* = recall
- *P* = precision
- *H* = harmonic mean

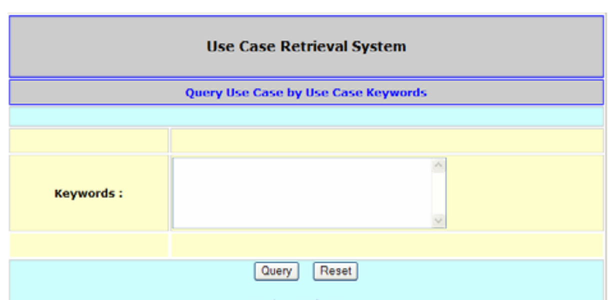


Figure 3. Use case query by use case keywords

Table 1. Average recall, precision, and harmonic mean of query by (1) use case keywords, (2) all use case keywords in each use case component, and (3) use case structure

Query by use case keywords (1)				Query by use case keywords (2)			Query by use case structure (3)			*P*+ of (3) from (1)	*P*+ of (3) from (2)
Subject	*R*	*P*	*H*	*R*	*P*	*H*	*R*	*P*	*H*		
A	0.2290	<u>0.4791</u>	0.2725	0.5147	<u>0.5565</u>	0.4915	0.4075	<u>0.6945</u>	0.4781	44.96%	24.80%
B	0.2789	<u>0.1863</u>	0.2032	0.4922	<u>0.2168</u>	0.2895	0.3180	<u>0.2559</u>	0.2554	37.36%	18.04%
C	0.6054	<u>0.1162</u>	0.1887	0.7232	<u>0.1094</u>	0.1859	0.7429	<u>0.2785</u>	0.3519	139.67%	154.57%
D	0.3490	<u>0.3073</u>	0.2863	0.5865	<u>0.3322</u>	0.3937	0.4615	<u>0.4956</u>	0.4344	61.28%	49.19%
E	0.6125	<u>0.1831</u>	0.2717	0.7125	<u>0.1590</u>	0.2525	0.5438	<u>0.2572</u>	0.3237	35.01%	61.76%
Average	0.4149	<u>0.2544</u>	0.2445	0.6058	<u>0.2748</u>	0.3226	0.4947	<u>0.3963</u>	0.3687	55.78%	44.21%

From table 1, recall, precision, and harmonic mean indicate that recall, precision and harmonic mean of use case query by use case structure is higher than those of use case query by use case keywords for almost all sample subjects. Average recall, precision, and harmonic mean of all subjects also indicate that recall, precision and harmonic mean of use case query by use case structure is higher than those of use case query by keywords. According to precision value, the precision improvement (*P*+) is in a high rate for all sample subjects. Finally, the average *P*+ of (3) from (1) and *P*+ of (3) from (2) are concluded to be 55.78% and 44.21%. The average recall, precision, and harmonic mean are shown in figure 4.

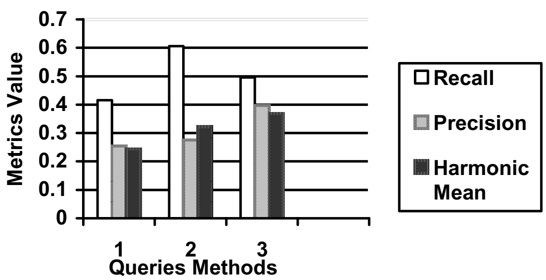


Figure 4. Graph represents the average recall, precision, and harmonic mean from table 1.

According to the experimental results, statistical analysis is used to support reliability of them. Z-test for hypothesis testing is applied to our experimental results.

The hypothesis of the experiment is that use case retrieval by use case structure is more effective than use case retrieval by use case keywords. Three selected metrics which are recall, precision, and harmonic mean are used to measure effectiveness of all use case retrieval systems. Therefore, H_0 and H_1 is defined as follows.

$$H_0 = \overline{r_1} - \overline{r_2} = 0 \quad H_0 = \overline{p_1} - \overline{p_2} = 0 \quad H_0 = \overline{h_1} - \overline{h_2} = 0$$
$$H_1 = \overline{r_1} - \overline{r_2} > 0 \quad H_1 = \overline{p_1} - \overline{p_2} > 0 \quad H_1 = \overline{h_1} - \overline{h_2} > 0$$

Where:

$\overline{r_1}$, $\overline{p_1}$, and $\overline{h_1}$ = average recall, precision, and harmonic mean of use case retrieval by use case structure.

$\overline{r_2}$, $\overline{p_2}$, and $\overline{h_2}$ = average recall, precision, and harmonic mean of use case retrieval by keywords (1).

The significance value is defined to be 0.05, so confidence value is 0.95. Therefore, the hypothesis H_0 will be rejected when $z > z_{.95}$, and $z_{.95}$ is 1.645 referred from z table [19]. The results of z-test of the experiments are shown in table 2.

From summary weighted value of query by use case structure in table 3, query is the number of queries in a component in which users generate keywords. The components whose query is more than average query are regarded as the important components.

From table 3, it can be observed that use case name, objective, and actor are three components whose query is more than average. Therefore, use case description template consisting of three components shown in figure5

Table 2. The summary results of z-test of recall, precision, and harmonic mean of the experiment

	Topic A	Topic B	Topic C	Topic D	Topic E	Average
Recall	3.6046	<u>1.1512</u>	2.2765	1.9698	<u>-1.7343</u>	<u>1.3834</u>
Precision	3.6075	2.3498	3.6218	3.2639	3.1068	2.5076
Harmonic mean	4.2498	2.3926	4.2025	2.9908	2.0910	3.0594

may be an effective template for use case retrieval. Meanwhile, average weighted value defined by users is also significant. The components whose average weighted value is more than the average are also regarded as the important components. It can be observed that use case name, objective, actor, association, precondition, postcondition, and normal flow of events are seven components whose average weighted value is more than average. Therefore, use case description template consisting of seven components shown in figure 6 may also be an effective template for use case retrieval.

Three components use case description template
Use Case Name :
Objective :
Actor :

Figure 5. Use case description with three components

Seven components use case description template	
Use Case Name :	
Objective :	
Actor :	
Association :	
Precondition :	
Postcondition :	
Normal Flow of Events :	

Figure 6. Use case description with seven components

From use case description template with three components in figure 5 and with seven components in figure 6, the experiment is tested again in order to find effectiveness of use case retrieval using both use case description templates. Their results are shown in table 4. From the results, we can conclude that the components of a use case description have an effect on use case retrieval results. Components that the users always generated and defined high weighted value are regarded as the important components of a use case description.

According to the precision of use case retrieval in table 4, the results of use case retrieval considering only important components are more precise than those of use case retrieval considering all components of a use case description. The precision improvement ($P+$) of (2) from (1) and of (3) from (1) are concluded to be 0.66% and 3.46%. It can be observed that $P+$ of (3) from (1) is higher than $P+$ of (2) from (1). Therefore, we can conclude that a use case description template which is efficient for use case retrieval is composed of seven components shown in figure 6.

Table 3. Summary of weighted value of user's query by use case structure

Component	Query (0-200)	Average weighted value (1-5)
Use case name	197	3.89
Objective	143	3.54
Actor	101	3.63
Association	7	3.57
Include	0	1.00
Extend	2	2.50
Generalization	0	1.00
Precondition	9	2.89
Postcondition	5	4.00
Normal flow of events	7	2.71
Subflow of events	1	2.00
Alternative flow of events	0	1.00
Average	39.3333	2.6442

6. Summary and Future Work

This paper proposed an approach for retrieving use cases using use case terms and structure similarity computation. Information storage and retrieval theories such as automatic indexing, term weighting system, and similarity computation are applied to storage and retrieval process of this approach. In order to measure effectiveness of the approach, the tool supporting use case retrieval is developed, and the experiment is designed to compare effectiveness of use case retrieval by use case keywords, which is a general approach to use case retrieval by use case structure, which is our approach. The experiment has 10 testers to test both use case retrieval systems. Many factors such as use case collection, users, search process, and queries are controlled to reduce the amount of bias in the experiment. Three selected metrics which are recall, precision, and harmonic mean are used to evaluate the effectiveness of both use case retrieval systems. According to the statistical result analysis shown in table 2, although recall is not assured that use case retrieval by use case structure is more effective than use case retrieval by use case keywords, precision and harmonic mean, which is the metric combining recall and precision, can conclude that use case retrieval by use case structure is more effective than use case retrieval by use case keywords. Therefore, they can be concluded that use case retrieval by use case structure which is our approach

Table 4. Average recall, precision, and harmonic mean of (1) query by use case structure, (2) query using three components, and (3) query using seven components

Query using all components (1)				Query using 3 components (2)			Query using 7 components (3)			*P+* of (2) from (1)	*P+* of (3) from (1)
Subject	*R*	*P*	*H*	*R*	*P*	*H*	*R*	*P*	*H*		
A	0.4075	0.6945	0.4781	0.4083	0.6991	0.4799	0.3980	0.7113	0.4785	0.66%	2.42%
B	0.3180	0.2559	0.2554	0.3125	0.2537	0.2529	0.2938	0.2570	0.2511	-0.86%	0.43%
C	0.7429	0.2785	0.3519	0.7375	0.2796	0.3520	0.6875	0.2939	0.3637	0.39%	5.53%
D	0.4615	0.4956	0.4344	0.4635	0.4993	0.4371	0.4604	0.5165	0.4486	0.75%	4.22%
E	0.5438	0.2572	0.3237	0.5438	0.2626	0.3263	0.5188	0.2714	0.3344	2.10%	5.52%
Average	0.4947	0.3963	0.3687	0.4931	0.3989	0.3696	0.4717	0.4100	0.3753	0.66%	3.46%

is more effective than use case retrieval by use case keywords.

However, the approach still has some limitations. Firstly, the retrieval process can consider only similarity of word, but it cannot consider similarity of semantic. Therefore, average recall may become lower if the user queries use cases by generating keywords whose semantic is similar to use case in the collection, but those keywords do not appear in the collection. Another limitation is weighted value generated by the users. If the users select poor weighted value, average recall might become lower too. For improving recall, correctness rate of retrieval system, this research can use thesaurus in the retrieval system. It helps the retrieval system consider similarity of semantic of use cases. Another information storage and retrieval theory, user relevance feedback, can be applied for improving weighted value of the users. These may improve recall of use case retrieval system. Furthermore, probabilistic model, and document clustering are also included in the future work for improving effectiveness and efficiency of the use case retrieval system.

7. References

[1] Professional Practices Committee, Software Engineering Body of Knowledge, IEEE, 2004.

[2] C. Mcclure, Software reuse techniques, 1 ed, Prentice Hall, 1997.

[3] C. W. Krueger, "Software reuse," ACM Computing Surveys, vol. 24, pp. 131-183, 1992.

[4] J. Rambaugh, G. Booch, and I. Jacobson, The unified modeling language: The user guide, Addison-Wesley, 1998.

[5] H. Gomaa, Designing concurrent, distributed, and real-time applications with UML, Addison-Wesley, 2000.

[6] E. J. Braude, Software Engineering: An Object-Oriented Perspective, 1 ed, John Wiley & Sons, Inc., 2001.

[7] M. C. Blok, and J. L. Cybulski, "Reusing UML specifications in a constrained application domain," Asia Pacific Software Engineering Conference, 1998.

[8] M. Saeki, "Reusing use case descriptions for requirements specification: towards use case patterns," Asia Pacific Software Engineering Conference, 1999.

[9] M. Saeki, "Patterns and aspects for use cases: reuse techniques for use case descriptions," Requirements Engineering Conference, 2000.

[10] F. Marir, K. Haouam, "Rhetorical Structure Theory for content-based indexing and retrieval of web documents," ITRE, 2004.

[11] H. G. Woo and W. N. Robinson, "Reuse of scenario specifications using an automated relational learner: a lightweight approach," Requirements Engineering Conference, 2002.

[12] M. F. Porter, "An algorithm for suffix stripping, Program," Automated library and information systems, vol. 14, pp. 130-137, 1980.

[13] R. Baeza-Yates and F. B. William, Information retrieval: Data structures and algorithms: Prentice-Hall, 1992.

[14] B. Wixom, A. Dennis, and D. Tegarden, Systems analysis and design: An object-oriented approach with UML, 1 ed, Wiley, 2002.

[15] G. Winters, "Use case terminology," Software, IEEE, vol. 22, pp. 67, 2005.

[16] M. J. McGill and G. Salton, Introduction to modern information retrieval, McGraw-Hill, 1983.

[17] B. Ribeiro-Neto and R. Baeza-Yates, Modern information retrieval, Addison-Wesley, 1999.

[18] http://www.ets.org/toefl

[19] G. Keller and B. Warrack, Statistics for Management and Economics, Thomson, 2003.

0-7695-2685-3/06 $20.00 © 2006 IEEE