RETRIEVING USE CASE DIAGRAM WITH CASE-BASED REASONING APPROACH

[1]B. SRISURA, [2]Dr. J. DAENGDEJ

[1,2]Autonomous System Laboratory, Science and Technology Faculty, Assumption University, Thailand

ABSTRACT

Due to the difficulty of recalling the semi-structure form of use case diagram, applying reusing approach to the requirement analysis process has been being concerned as a complex task. Despite of this fact, there are relatively little attentions found in the research community. Therefore, this paper proposes how Case-Based Reasoning (CBR) – an approach solves new problem from recalling experiences, can be effectively applied to support the use case diagram reuse. In this research, the diagram retrieval method is designed to match the use case diagram by considering two dimensions: use case and actor dimension and relationship dimension. However, in order to present the significant accuracy and practicality of the proposed method, a tool and five comparative sets– including four various dimensional weights and one commercial tool were carefully set up to test in an experiment.

Keywords: *Diagram-Based Retrieval, Use Case Diagram Retrieval, Use case Reuse*

1 INTRODUCTION

While the use case diagram - the most common used artifact in the requirement analysis stage of the software development life cycle, plays the main role in gathering the software requirements [1], [2], reusing use case diagram is proposed to help analysts to model their use case diagrams in a short period. Since the semi-structure form of use case diagram – including both text-based and structure-based format, the practicality of reusing is concerned as a complicated task which encourages researchers to resolve.

Interestingly, most of analysts are familiar with modeling a use case diagram through recalling their own experiences from the previous works. The more experiences analysts have, the better they can perform a modeling job. Modeling use case diagram based on the experiences enables analysts to reuse their previous works rather than stating from scratch. Like Case-Based Reasoning (CBR) [3], [4], it is a process model of applying prior experience to a new problem to avoid the time necessary to create a solution from scratch. Therefore, CBR can be an attractive approach being applied in developing a retrieval aid to support the use case diagram reuse.

There are two CBR sub-processes that must be addressed in this paper: organizing case base – the use case diagram base, and retrieving the most relevant cases – the use case diagram retrieval method, to the new problem. The proposed retrieval method retrieves the similar diagrams by matching two dimensions: use case and actor dimension – text-based format, and relationship dimension – structure-based format, and then ranking them before proposing to analysts. The matching score and reasonable weight of each dimension are carefully calculated based on the nearest neighbor matching and ranking to deliver the most similar use case diagram in order.

Moreover, an experiment is implemented to evaluate the proposed diagram retrieval method performance and ensure that it can be used to effectively retrieve the most similar use case diagram from a use case based - a repository of well- defined use case diagram from previous works, rather than using the keyword search.

In order to clarify the big picture of designing the proposed diagram retrieval method, an introduction and related works are mentioned in the first and second section, respectively. In constructing a case-based retrieval method, measured features and their rational weights used in matching function are identified in Section 3. Section 4, therefore, illustrates the proposed case-based retrieval architecture in details. Section 5 describes the retrieval method through an example. Then, four comparative feature weights and one commercial tool - shown in section 6, are set up to evaluate the acceptable retrieval performance. Finally, conclusion and recommended future research are summarized in the last section.

2 RELATED WORKS

Various attempts of researchers, in proposing use case retrieval methods, have been being run, continuously. Most research works proposed their retrieval methods by focusing at the explanation of use case diagram, such as use case description, activity diagram, or sequence diagram.

The first attempt, when a use case diagram is explained with a sequence diagram in the stage of realizing use case diagram, the use case retrieval indirectly evaluates use case diagram via the sequence diagram similarity. Block and Cybulski [5] initialize the first retrieval use case method which calculates the sequence diagram similarity from the diagram event flow accumulation. Ten year later, Woo and Robinson [6] proposes an automated technique in calculating the sequence diagram similarity. The sequence diagram is transformed into graph and then graph-based matching is activated for measuring the diagram similarity.

In case of explaining a use case diagram with the use case description, most of retrieval methods in the second attempt rely solely on the Information Retrieval (IR) approach [7] because the description is represented as text-based. In IR research community, Udomchaiporn, Prompoon and Kanongchaiyos [8] propose Vector Space Model (VSM) retrieval method to measure use case description similarity. Moreover, Suksaard and Prompoon [9] also proposed an extended idea to improve Udomchiporn's use case retrieval method with the relevance feedback technique.

According to the reviewed research works, most use case retrieval methods are properly worked when the explanation of a use case diagrams, which can be either use case description or sequence diagram, is already specified. However, in the requirement analysis stage which is the situation that this paper focuses, analysts have only a use case diagram – excluding any explanations. The explanation of use case diagram is actually implemented in analysis and design stage. Therefore, retrieving use case by considering its explanations can not be occupied in the requirement analysis situation. According to this reason, the diagram retrieval method, which considers only a use case diagram itself, is required for supporting analysts to reuse use case diagrams in the requirement analysis stage.

There is a growing interest to apply CBR approach for building knowledge-based systems over two decades. Several CBR systems have been built to assist human designer to find appropriate design cases: ARCHIE [10] are CBR tools for aiding designer during conceptual design in architecture, CASECAD [11] is a multimedia case-based assistant in the structural design building. In the software development process, Case-Based Reasoning (CBR) can be successfully applied in reusing artifacts of the design stage, such as class diagram [12]-[14], which are less abstract than the requirement analysis stage. Due to the document-based and semi-structure form of artifacts found in the requirement analysis stage, approaching CBR in this stage has been being questionable idea and encouraged to be researched.

Although the semi-structure form of use case diagram is the main problem issue, this paper presents that CBR approach has enough potential to be applied for reusing the use case diagram in the requirement analysis stage, effectively.

3 DIMENSIONAL MATCHING FUNCTION

In the retrieval process of CBR, dimensions and their related weights are firstly requested to be identified for matching query case with similar cases [3]. Therefore, when the concerned case is use case diagram, important dimensions and their rational weights are necessarily be defined in this section.

In determining use case diagram dimension, since the use case diagram – query diagram, is used to be searched; the diagram components should be used to determine its similarity. Generally, a use case diagram consists of four components: actor, use case, relationship, and system boundary [15]. These components are specified to represent all participated actors and functional requirements of a new system therefore they are proposed to be used in matching the relevant use case diagrams from a use case base.

Considering four components, they have some similar characteristics which can be combined to optimize the matching time. Therefore, this research classifies them into two dimensions based on their characteristics. Firstly, the use case and actor dimension – including use case, actor and system boundary components, are usually represented as text-based information. The second dimension is relationship dimension represented as structure-based information which can be specified into three sub-components: relationship type, navigator and multiplicity.

After defining what dimensions are suitable to be matched, a common numerical evaluation function used in CBR, the nearest-neighbor matching [3] is applied to compute the aggregate degree of matching along with a particular dimension and its importance (or weight) as follows:

$$S(Q,D) = \frac{\sum_{i=1}^{n} w_i \times sim(d_i^Q, d_i^D)}{\sum_{i=1}^{2} w_i}; w_i \in \Re = [0,1] \quad (1)$$

Given: *S(Q,D)* is a use case diagram matching score between a query use case diagram (*Q*) and a concerned use case diagram (*D*) along two dimensions (*n=2*). And, w_i is the importance of each dimension. Finally, $sim(d^Q_i, d^D_i)$ is the representative of the similarity matching scores under each dimension which is described in details as the following:

3.1 Use Case and Actor Dimensional Matching

Due to the system boundary, use case and actor names are represented as text-based format which can be either single word, such as actor name "Customer" or phase, such as use case name "Order product". Most of partial matching IR models [16], such as Vector Space Model (VSM), Probabilistic Model (PM), determine document similarity from considering word occurrence (or term frequency). A shorten unique name is recommended to define use case and actor name. Therefore, term frequency is ineffective factor to evaluate use case diagram similarity. The proposed the similarity matching score under use case and actor dimension can be a fairly straightforward function as follows:

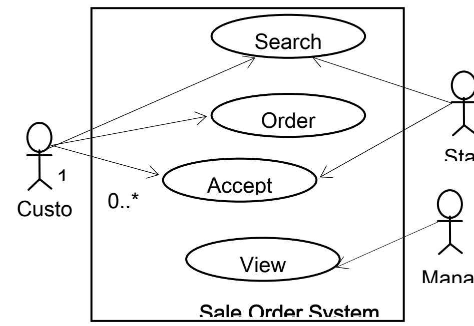
When n is the total number of matched words found in the considered use case diagram and m is the total query diagram matched word number,

$$sim(d_1^Q, d_1^D) = \frac{\sum_{i=1}^{n} 1}{\sum_{j=1}^{m} 1} \quad (2)$$

3.2 Relationship Dimensional Matching

In relationship dimension, three sub-components: relationship type, navigator and multiplicity, are considered to measure entire relationship found in a use case diagram. Although relationship in a diagram can be viewed as edge in a graph, some situation use case diagram is drawn as the disjoint graph shown in Figure 1. Use case diagram can not actually be represented as tree structure because some use case, such as "Search product" in Figure 1, may have several predecessors. Therefore, graph-based or tree-based similarity measurement can not be applied to evaluate use case diagram similarity.

Figure 1. Use case diagram



exa

mple (Not graph and tree)

From the given supported reasons, semi-structure-based of use case diagram similarity is simplified component by component. All relationships around each use case and actor are summarized in average to evaluate the similarity matching score of this dimension as follows:

$$sim(d_2^Q, d_2^D) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} w_R(V_{i,j}^R) + w_N(V_{i,j}^N) + w_M(V_{i,j}^M)}{n \times m} \quad (3)$$

When: $V^R_{i,j}$ is relationship type value of relationship j in use case i, $V^N_{i,j}$ represents navigator value of relationship j in use case i, $V^M_{i,j}$ is defined as multiplicity value of relationship j in use case j, m is the total number of relationships around use case i and n is the total number of use cases and actors in use case i. Unfortunately, weights of relationship sub-components are required to specify their importance as the follows:

3.2.1 Sub-component weight

Without user defined weight setting, default weights of three sub-components - type, navigator, and multiplicity, are automatically activated by based on the utilized criteria summarized as follows:

Table 1. Sub-component default weight

Sub-component	Notation	Default Weight
Relationship Type	w_R	0.5
Navigator	w_N	0.3
Multiplicity	w_M	0.2

3.2.2 Sub-component value

3.2.2.1 Relationship type value ($V^R_{i,j}$)

Considering from the important role of each relationship defined in [17], these values can be summarized in Table 2.

Table 2. Relationship type value

Relationship Type	Default Value ($V^R_{i,j}$)
Association	0.4
Generalization	0.1
Include	0.3
Extend	0.2

3.2.2.2 Navigator value ($V^N_{i,j}$)

Considering from the potential roles in setting the relationship navigator [17], it is classified into two types: primary and secondary actor. The primary navigator is usually set as subject whose priority is considered as the top concern. Therefore, the priority of default weight value of subject navigator is higher than the object navigator. The default navigator values are defined as follows:

Table 3. Navigator value

Navigator	Default Value ($V^N_{i,j}$)
Subject (→)	0.7
Object (←)	0.3

3.2.2.3 Multiplicity value ($V^M_{i,j}$)

Like relationship value assignment, the multiplicity value can be summarized in Table 4.

Table 4. Multiplicity value

Notation	Multiplicity	Default Value ($V^M_{i,j}$)
1	Be only one	0.1
0..1	Must be 0 or 1	0.2
1..*	As few as 1	0.4
0..*	Optional	0.3

4 CASE-BASED REASONING RETRIEVAL ARCHITECTURE

4.1 Diagram Case Representation

Currently, the concerned case is a use case diagram therefore the used format in case based – the use case base, is organized by concerning its compatibility. Since a use case diagram being collected to the use case base might be generated from various modeling tools which represent use case diagram with their unique format, most of software design tools provide an option for exporting diagrams into a XMI (eXtended Mark-up Language) format.

An XMI use case diagram example shown in Figure 2 is automatically generated from a software modeler – IBM Rational Software Modeler V.7.0.5 which will be stored in a use case based [18]. Therefore, a considered case base is represented as XMI use case base.

```
<packagedElement xmi:type="uml:Actor"
xmi:id="U66529ddb" xmi:uuid="66529ddb"
name="Staff" visibility="public">
        <generalization
        xmi:type="uml:Generalization"
        xmi:id="U50975722" xmi:uuid="50975722"
        general="U7f075f94"/>
</packagedElement>
```

Figure 2. XMI Use case diagram representation

4.2 Diagram Retrieval Method

Based on CBR approach, use case diagram retrieval process is the combination of weighting, matching and ranking. Choosing the most similar case is primarily partial-matching process when searching calls a matching function to compute the degree of match along each concerned dimension. The architecture of the proposed method is presented in Figure 3.

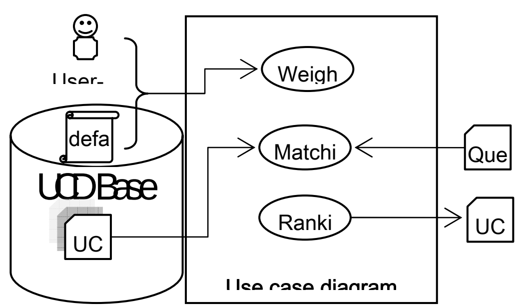


Figure 3. Use case diagram retrieval architecture

4.2.1 Weighting module

This module allows user to determine their diagram component weights. However, without user defined weight setting, this module will invoke the default weight values discussed in Section 3.

4.2.2 Matching module

Two dimensions – use case and actor dimensional matching and relationship dimensional matching, are constantly defined based on the important of each dimension which are prioritized by using the discrimination network weighting scheme [3]. In Figure 3, the prioritized dimensional matching sequence is illustrated as follows:

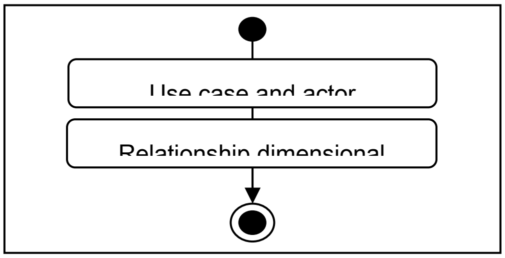


Figure 4. Prioritized dimensional matching scheme

4.2.3 Ranking module

To provide the flexibility in considering the suitable degree of matching, ranking process is provided for supporting user to select the use case diagrams, easily. Various numbers of the retrieved cases are represented in term of the degree of matching found (D_m %) and the degree of the required matching (D_r %). Given: N is the total number of searched words which must be matched in the concerned case when we compare to the entire matched words in a query, the degree of matching found (D_m %) is defined as:

$$Dm = \frac{\sum_{i=1}^{n} 1}{N} \qquad (4)$$

When m is the total use case diagram case whose D_m is specified and M is the total use case diagram case which has a word to be matched, the degree of the required matching (D_r %) can be calculated as:

$$D_r = \frac{\sum_{j=1}^{m} 1}{M} \qquad (5)$$

5 RETRIEVAL EXAMPLE

In order to see how each module of the proposed diagram retrieval works, an example is presented in this section.

Given a query use case diagram and four use case diagram examples shown in Figure 5 - Figure 9,

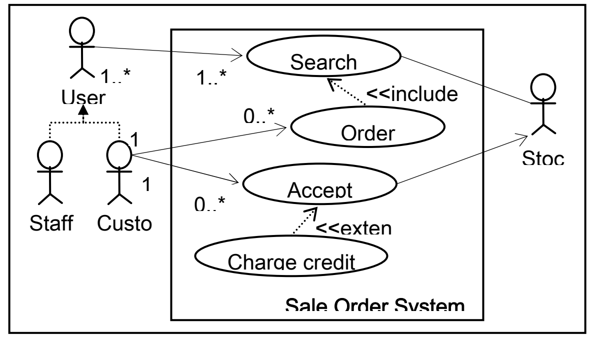


Figure 5. Use case diagram query

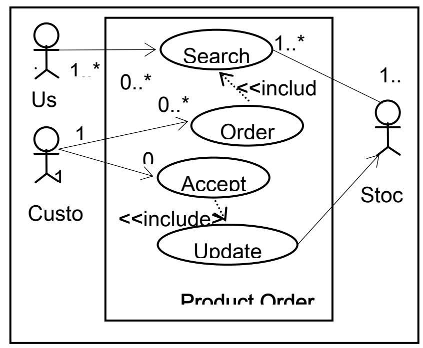


Figure 6. Diagram case 1

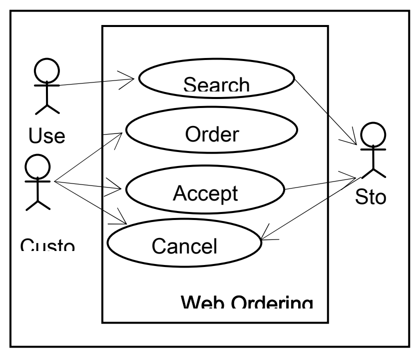


Figure 7. Diagram case 2

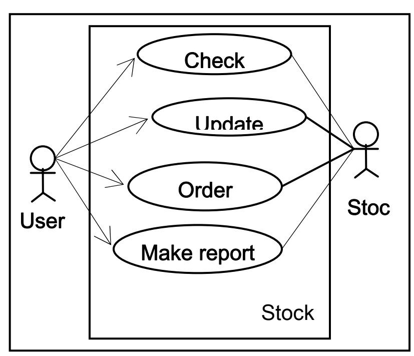


Figure 8. Diagram case 3

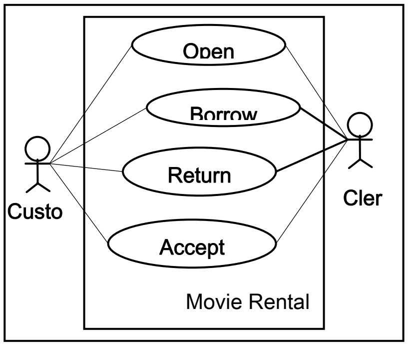


Figure 9. Diagram case 4

Firstly, the entire words, found in each system boundary, use case and actor, will be extracted – such as a use case "search product" is extracted as two words (search and product) and then searched in the provided dictionary table which is a word collection. Word search capability has yet limited at exact matching and will be researched with the reasonable approach in advance. More complex diagram query is needed is more time-consumed word search. Therefore, clustering use case is provided to group any related use case diagrams which have similar words to support search process with a reasonable time as shown in Figure 10.

Word	UCD Lists
card	-
charge	-
credit	-
custo	1, 2, 4
	……

Figure 10. Use case clustering in dictionary

Next, computing the similar matching score under the use case and actor dimension can be performed by using formula (2). From the given examples, repeating word in a diagram – such as "product" in the query diagram, "stock" in case 3 and "movie" in case 4, are count as one time because word frequency factor has not yet been used to judge the diagram similarity and will be research further. From the given query, in Figure 5, all extracted words are summarized and shown below.

Table 5. Use case and actor dimensional matching

Query Word	Use case diagram case						Query
	1	*2*	*3*	*4*	*5*	*6*	
accept	1	1	-	1	1	-	1
card	-	-	-	-	-	-	1
charge	-	-	-	-	-	-	1
credit	-	-	-	-	-	-	1
customer	1	1	-	1	-	-	1
order	1	1	1	-	-	-	1
payment	-	1		1	1		1
product	-	1	1	-	-	-	1
search	1	1	-	-	-	-	1
staff	-	-	-	-	-	1	1
stock	1	1	1	-	-	-	1
user	1	1	1	-	-	-	1
Total Matched	**6**	**8**	**4**	**3**	**2**	**1**	**12**
sim **(d_1^Q,d_1^D)**	**0.50**	**0.67**	**0.33**	**0.25**	**0.17**	**0.08**	**1**

All retrieved cases will be ranked based on degree of matching found (D_m %); however filtering use case diagram task can be done with two provided option including user judgment and the default option is automatically generated by choosing the minimum D_r % to be the default D_r %. Only one word is required to be found, this diagram case will be retrieve to check with the next dimension. From the given example ranking and filtering process are summarized in Table 6. Suppose a user choose three candidate diagrams, the use case diagram case 1, 2 and 3 are retrieved to process the relationship dimensional matching, further. However, without user setting, the default ranking is defined at D_m % > 0. Therefore, the use case diagram case 1-6 are automatically be retrieved.

Table 6. Ranking of similarity matching score under use case and actor dimensional matching

D_m %	Matched (n)	Use case diagram (UCD_i)						Retrieved (m)	D_r %
		2	1	3	4	5	6		
100	12							0	0
92	11							0	0
83	10							0	0
75	9							0	0
67	8	0.67						1	17
58	7	0.67						1	17
50	6	0.67	0.50					2	33
42	5	0.67	0.50					3	50
33	4	0.67	0.50	0.33				3	50
25	3	0.67	0.50	0.33	0.25			4	67
17	2	0.67	0.50	0.33	0.25	0.17		5	83
8	1	0.67	0.50	0.33	0.25	0.17	0.08	6	100

To compute the matching score under relationship dimensional matching, the similarity matching score of the query and entire filtered use case diagrams (UCD_1 – UCD_6) from the previous step are calculated. Given a use case example named "Search" of use case diagram case 1, in Figure 6, recall formula (3) in Section 3 which can be shown in Figure 11.

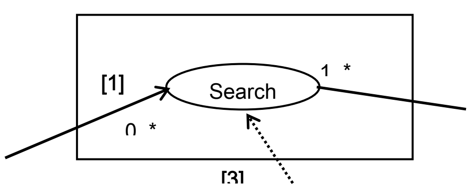


Figure 11. Relationship around use case "Search"

Refer to use case diagram case 1, there are seven use cases and actors, therefore *n* value is 7. When recalling the relationship sub-component weights and values defined in Section 3, the similarity matching score under relationship dimension of the use case diagram case 1 can be presented in term of formula (3) as follows:

$$sim(d_2^Q, d_2^D) = \frac{\sum_{i=1}^{7}\sum_{j=1}^{m} 0.5(V_{i,j}^R) + 0.3(V_{i,j}^N) + 0.2(V_{i,j}^M)}{7m}$$

Therefore, formula (1) is activated to calculate the use case diagram similarity. Suppose that w1 is 0.6 and w2 is 0.4, then the diagram similarity of use case diagram case 1 is defined as follows:

$$S(Q, D_1) = 0.6(0.5) + 0.4(0.8) = 0.62$$

Table 7, six candidate diagram similarities are automatically ranked as follows.

Table 7. Use case diagram similarity ranking

$S(Q,D_i)$					
1	2	3	4	5	6
0.62	0.60	0.36	0.27	0.18	0.13

From the highlight row in Table 7, it is implied that if users accept the degree of matching 60% the use case diagram case 1, whose use cases, actors and relationship are mostly matched to the query use case diagram. However, users may adjust the candidate use case diagrams to fit to your requirement by changing the dimension weights.

6 EXPERIMENTAL EVALUATION

An experimental was performed with a supporting tool which is implemented to evaluate the proposed diagram retrieval method performance in term of its accuracy with these measurements: recall, precision and F-measure [7], [16]. However, the experiments were set up as the followings:

6.1 A Supporting Tool

User can input a query by importing a use case diagram data in XMI format. Before retrieving module is executed, users are requested to set the dimension weight through a provided menu. However without user defined weights, the default weight, mentioned in Section 3, will be activated in retrieving similar use case diagram instead. Figure 12 shows those mentioned options.

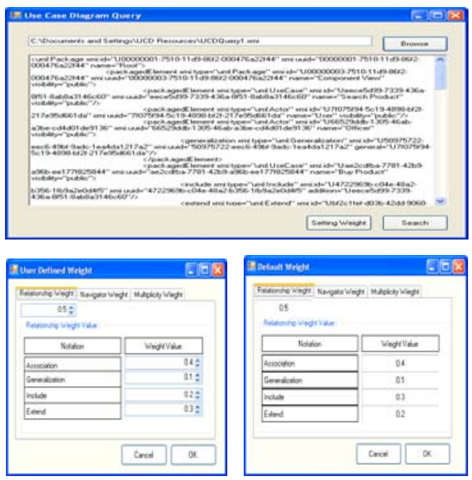


Figure 12. Use case query, User defined and default weight setting

The matching module is automatically generated when user clicks the search button. Then the degree of matching found *(D_m %)* and the degree of the required matching *(D_r %)* are reported to the users to consider and select the most similar use case diagram by themselves. Figure 13 shows that the use case retrieval tool requests the degree of matching found from users for using in ranking module.

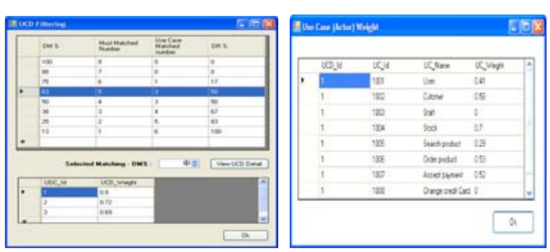


Figure 13. Use case diagram ranking

6.2 Diagram Collection

The previous designed use case diagrams were collected mostly from academic projects and examples from text books (around 120 use case diagrams from 4 main various domains: inventory management system, sale ordering system, educational information system and hospital management system). However, the numbers of use case and actor in each diagram are approximately 5 to 30 use cases.

6.3 Retrieval Experiment

Since the most common use case retrieval methods used to compare in the most reviewed papers is keyword search based on IR approach, it is also implemented to compare with their proposed diagram retrieval method. Because the proposed retrieval method is the diagram retrieval – under the limited constraint that the diagram explanations have not yet been identified, use case retrieval method through the use case explanation mentioned in the reviewed research cannot be used to compare, appropriately.

In CBR view, the keyword search is implemented in the way of performing use case and actor dimensional matching, only.

Twenty problems were generated from the use case base. From four application domains, five use case diagrams were randomly generated from each application domain. For each problem, a set of relevant diagrams and the most similar one were prepared to be used to justify the proposed retrieval method.

However, the reasonable numbers of comparative sets were considerably inserted to test the focused retrieval performance rather than comparing with the keyword search. According to the proposed two dimensions: use case and actor dimension and relationship dimension, four variations of the dimension weights were designed to test the proposed retrieval performance shown in Table 8.

Table 8. Four comparative sets

Set	Use case and Actor dimensional Weight (w_1)	Relationship dimensional weight (w_2)
W_1	1	0
W_2	0.6	0.4
W_3	0.4	0.6
W_4	0	1

Obviously, W_1 and W_4 represent an idea of considering only use case and actor dimensional matching – a representative of keyword search, and relationship dimensional matching, respectively. Another effort to consider each dimension important, therefore W_2 – the use case and actor is more important feature than relationship, and W_3 – the relationship is more important feature than use case and actor. These weights can be set into five comparative sets as follows:

- W_1: The retrieval method which considers only use case and actor dimension.
- W_4: The retrieval method which considers only relationship dimension.
- W_2: The retrieval method which considers both use case and actor dimension and relationship dimension with the different ratio, however the use case and actor dimension importance is higher than the relationship dimension (w_1:w_2 = 0.6:0.4).
- W_3: The retrieval method which considers both use case and actor dimension and relationship dimension with the different ratio which the relationship dimension importance is higher than the use case and actor dimension (w_1:w_2 = 0.4:0.6).
- W_5: IBM Rational Software Modeler V.7.0.5 is the commercial keyword search. Due to the difficulty of finding the suitable comparative set, reviewing scope is considered two tracks: in research community (academic track) and in commercial UML software modeling tools (industrial track).

6.4 Measurement

In each problem, the proposed retrieval technique retrieved a set of cases. Testing for sets of different sizes is required to study recall and precision values [16] along sets of these sizes which are defined as the following: 3, 5, 10 and 20.

Because recall and precision are antagonistic measures, F-measure – proposed by [7] to combine recall and precision in a single measure, is recommended to make an evaluation of the experiments.

$$F = \frac{2 \bullet \mathrm{Re}\,call \bullet \mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,ecision} \qquad (6)$$

Given: Recall is the number of retrieved relevant cases divided by the total number of relevant diagram in the use case base. Precision is the total number of retrieved relevant cases divided by the total numbers of retrieved cases.

The F-Measure average from 20 problem diagram queries in the different sizes: 3, 5, 10, 20 were used to calculate recall and precision and summarized in Table 9.

Table 9. Recall, Precision and F-Measure Results

	3			5			10			20		
	Recall	**Precision**	**F**	**Recall**	**Precision**	**F**	**Recall**	**Precision**	**F**	**Recall**	**Precision**	**F**
W_1	0.23	0.57	0.32	0.32	0.35	0.33	0.45	0.29	0.35	0.47	0.14	0.22
W	0.38	0.68	0.48	0.52	0.59	0.55	0.63	0.38	0.48	0.69	0.21	0.33
W_3	0.35	0.64	0.45	0.47	0.48	0.47	0.54	0.31	0.39	0.62	0.17	0.27
W_4	0.21	0.33	0.25	0.27	0.29	0.28	0.25	0.16	0.20	0.25	0.10	0.14
W_5	0.22	0.54	0.31	0.32	0.33	0.32	0.46	0.30	0.36	0.49	0.15	0.23

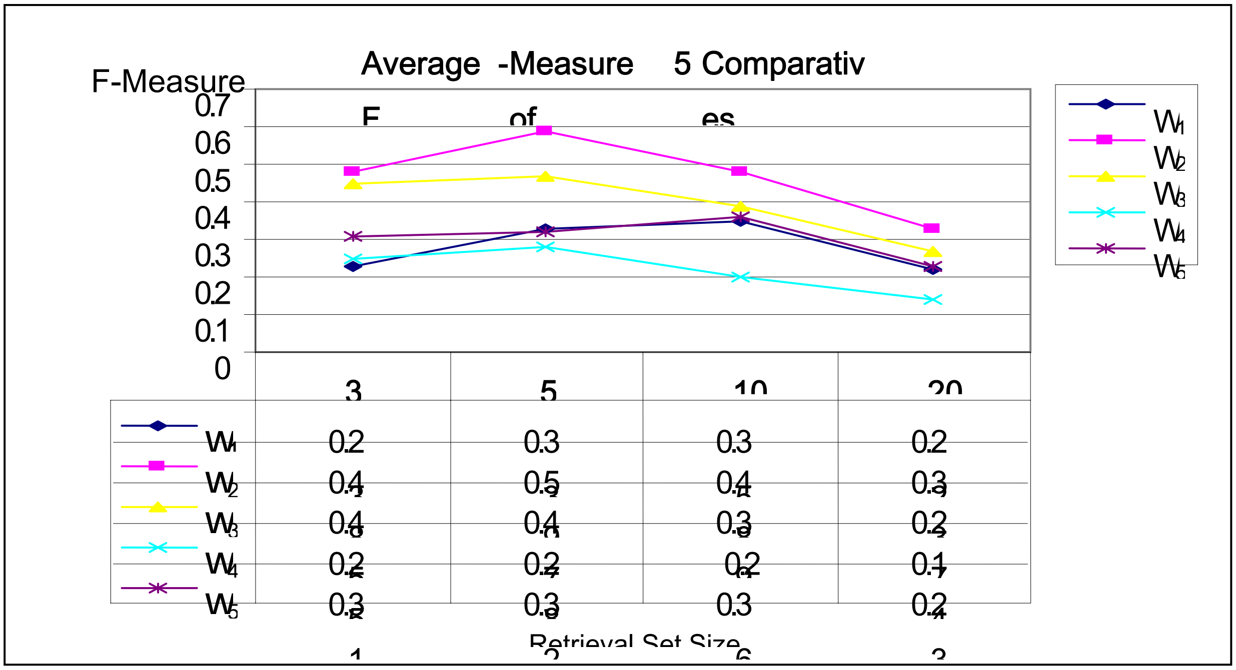


Figure 14. Average F-measure of five comparative sets

Usually, F-Measure value provides the quality of cases that are retrieved which can be used to evaluate the effectiveness of the proposed method. In order to assess the proposed retrieval performance, the average F-measure of five comparative sets (W_1 – W_5) were calculated and summarized in term of graph shown in Figure 14.

Considering the result in Figure 14, the comparative set W_2 (Using both dimensions to match the similar use case diagrams with the use case and actor dimensional weight, w_1 equal to 0.6 and the relationship dimension weight, w_2 is 0.4) and W_3 (Both dimensions are considered in matching but relationship dimensional weight is higher than use case and actor dimensional weight with the ratio w_2=0.4:w_1=0.6) have the most effective performances in retrieving the relevant use case diagram from use case base. Especially, W_2 (w_1=0.6: w_2=0.4) could be determined as a more suitable ratio than W_3.

Another couple comparative set (W_1 and W_5) their F-measure values are nearly closed because both techniques concern play the keyword search role. However, the average F-measure of W_5 (IBM Rational Modeler V.7.0.5 keyword search) is higher than W_1 (Only the use case and actor dimensional marching activation with the key word search). This result encourages us to further study how to improve the use case and actor dimensional matching of the proposed retrieval method in the next research. Finally, W_4 performance is determined as worst case. This can be interpreted that only relational dimension is not enough to be used to evaluate the use case diagram similarity.

7 CONCLUSION AND FUTURE RESEARCH

This research proposes a use case diagram retrieval method based on CBR approach to reuse previous designed use case diagram in the analysis and requirement stage of software development life cycle. Use case diagram components: system boundary, use case, actor and relationship, are used to determine the use case diagram similarity. Two dimensions – use case and actor dimension and relationship dimension, are proposed to design a suitable matching function of the retrieval method.

Three main modules of the retrieval method – weighting, matching and ranking, are designed to retrieve the most similar use case diagram, practically. In designing weighting module, it provides two main options: default weighting which determined from the experiment and user-defined weighting to support users in modifying their weights. In diagram matching, use case and actor dimension is designed to be previously matched to filter the reasonable use case diagram numbers before performing the relationship dimensional matching. After computing matching score, ranking process is designed to help users to select the most similar use case diagram, easily.

To reduce time and cost in modeling a use case diagram, retrieving the similar use case diagram by using the proposed retrieval method can help analyst to model a use case diagram, easier. However, there are some numbers of remaining issues required to be solved in future research.

First, from analyzing the experimental results; system boundary, use case and actor word matching is limited at exactly matching. Therefore, the "officer" and "staff" actor are judged to be different actor.

The next issue, improving matching algorithm performance, such as the user relevance feedback or suitable ratio of two dimension weights, is required to be researched.

Finally, the next stage of CBR process model, such as how to reuse use case diagram when the exact matched use case diagram number is not encountered or set of most similar use case diagrams are retrieved for enhancing the practicality of use case reuse in the real situation.

REFERENCES:

[1] I. Jacobson, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.

[2] P. Kruchten, *The Rational Unified Process 3rd Edition: An Introduction*, Addison-Wesley Longman Inc, 2004.

[3] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann, 2000.

[4] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundation Issues, Methodological Variations, and System Approaches", *Artificial Intelligence Communication (AICOM)*, IOS Press, Vol.7:1, 1994, pp. 39-59.

[5] M. C. Blok, and J. L. Bybulski, "Reusing UML Specifications in a Constrained Application Domain", *Proceeding of the 5th Asia Pacific Software Engineering Conference (APSEC)*, 1998.

[6] H. G. Woo, and W. N. Robinson, "A Light-Weight Approach to the reuse of use-cases specifications", *Requirement Engineering Conference (SAIS)*, 2002.

[7] C. J. V. Rijsbergen, *Information Retrieval*, The second edition, Butterworths, London, 1979.

[8] A. Udomchaiporn, N. Prompoon, and P. Kanongchaiyos, "Software Requirements Retrieval Using Use Case Terms and Structure Similarity Computation", *The 13th Conference in Software Engineering (APSEC)*, Dec 2006, p.113-120.

[9] S. Suksa-ard, and N. Prompoon, "Use Case Retrieval Using Use Case Clustering and User Relevance Feedback", *The 3rd Joint Conference on Computer Science of Software Engineering*, 2006.

[10] A. Goel, and J. L. Kolodner, "Toward a Case-Based Tool for Aiding Conceptual Design Problem Solving", *Proceedings of the DARPA Workshop on Case-Based Reasoning*, San Mateo, CA: Morgan Kaufmann, pp. 109-120.

[11] M. L. Maher, and M. B. Balachandran, "Flexible Retrieval Strategies for Case-Based Design", *Artificial Intelligent in Design '94*, Dordrecht: Kluwer Academic, pp. 163-180.

[12] I. Watson, S. Perera, "Case-Based Design: A Review and Analysis of Building Design Applications", *Journal of Artificial Intelligence for Engineering Design*, analysis and Manufacturing AIEDAM, 1997, Vol. 11, No.1, pp. 59-87.

[13] P. Gomes, F. C. Pereira, P. Paiva, N. Seco, P. Carreio, J. L. Ferreira, and C. Bento, "REBUILDER: A Case-Based Reasoning Design and Reuse Tool", *Journal of Engineering Intelligent System*, 2004.

[14] S. Channarukul, S. Charoenvikrom, and J. Daengdej, "Case-Based Reasoning for Software Design Reuse", *IEEE Aerospace Conference*, 2005.

[15] M. J. Chonoles, J. A. Schardt, *UML 2 for dummies*, Wiley Publishing, 2003.

[16] C. T. Meadow, *Text Information Retrieval Systems*, Academic Press, 1991.

[17] A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.

[18] M. Golbaz, A. Hasheminasab, and N. Daneshpour, "An XML Definition Language to Support Use Case-Based Requirements Engineering", *Proceeding of the international Multiconference of Engineers and Computer Scientists 2008(IMECS 2008)*, Vol. 1, 2008.