# A Storage and Retrieval of Requirement Model and Analysis Model for Software Product Line

Anavin Trakarnviroj and Nakhonthip Prompoon

*Abstract*—**There is different methods support the software reuse concept that helps project reduce the cost of software development. Software product line (SPL) is one of the widely used of the software reuse concept. This paper presents the concept of the storage and retrieval for software requirements model (SRM) and software analysis model (SAM) for SPL. The proposed concept is based on the principles of the PLUS which is used for the design of SPL. Applying the information storage and retrieval principle, the information relationships among all diagrams and documents of SRM and SAM components for SPL are established. A case study is also presented in order to illustrate the application of our concept.**

*Index Terms*—**Software Product Line, Requirement Model, Analysis Model, Information Storage and Retrieval**

## I. Introduction

Software reuse has been adopted in software engineering since 1968 [1]. The reuse goal is to support the developer to create a new system by using the previous software project assets have the similarity features. This leads to the reduction of software development cost. There are concepts and methods used in the software reuse such as software reuse libraries, software architecture and design reuse and software product line.

The field of software reuse has evolved from the reuse of individual components toward large-scale reuse with software product line (SPL) [1]. SPL integrates with software modeling approach that uses Unified Modeling Language (UML). It is called PLUS (Product Line UML-Based Software Engineering). The objective of PLUS is to explicitly model the commonality and variability in SPL. SPL consists of a group of small software components which can be used to create the new project components that have similar common features with the additional features. One example of the SPL is the software system embedded in the microwave oven. A microwave oven usually has different functions to provide the varieties of user cooking needs. The system can be divided into three characteristic features which include a common feature, an optional feature and an alternative feature. A common feature is a feature that all of the microwave oven must have such as heating system. An optional feature is a feature that some microwave oven has such as a turntable system. An alternative feature is a feature that each microwave oven has to provide additional services for a particular type of customer such as language display. All required outputs from SPL development life cycle should be documented and stored in a collection in such a way that can be easily used for the new project. In the case of microwave oven, the outputs such as documents, analysis and design model, source code and test cases may be classified into software development life cycle phase and software features. So the developer may use the concept of key word search to retrieve the match content of the SPL output from each phase to construct a new microwave oven model.

The concept of information storage and retrieval are widely used in various fields such as the popular search engines available in the software industry. This engine helps user search information that matches his/her interest based on key words appeared in the target document collection. This research propose a new concept for storing and retrieving the SPL components developed during the requirements identification and analysis phase based on the concept of vector space model in information storage and retrieval. The main focus of the SPL components are software requirements model (SRM) and software analysis model (SAM) mainly used UML to represent the system modeling. The models are use case model, feature model, static model and dynamic model. The proposed concept will help user retrieve the specific model by comparing the user interest presented in key word appeared in the query language and key word appeared in the SRM and SAM document. The match ones will be presented in a form that help user for further use according to the vector space similarity score ranking.

The rest of this paper is presented related works in section 2. Structure and modeling of SPL, types of documents used in the experiment of this research are presented in section 3. The approach of a storage and retrieval of requirement model and analysis model for SPL used in research and evaluation are described in section 4. Section 5 shows case study of this research. Section 6, we conclude our research and future work of this research.

## II. Related Work

Hussan Gomaa and Michael E. Shim [2] presented prototype of automated software product line engineering using multiple view model of software product line

Manuscript received January 12, 2012; revised January 12, 2012.

Anavin Trakarnviroj is with Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok Thailand (e-mail: anavin.t@student.chula.ac.th)

Nakhonthip Prompoon is with Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok Thailand (e-mail: nakhonthip.s@chula.ac.th)

architecture and developed component store in software product line repository. They developed product line may use UML to present the designed prototype.

Hanh Nhi TRAN, Bernard COULETTE and Dan Thu TRAN, My hang VU [12] proposed approach to reuse process knowledge. They presented a set of reuse operator that enable automatic applications of process pattern for generating and restructuring process models.

Paulo Gomes et al [3] proposed approach to reuse software using case-based reasoning and WordNet for storage and reusing design knowledge.

Markus Luckey et al [13] proposed extended quality model for reuse security requirements. Extended quality model is an extension of the activity-base quality model (ABQM) [14].

Akadej Udomchaiporn et al [4] proposed approach to retrieve software requirements specification using use case terms and similarity computation between key word appeared in each use case element and key word appeared in the stored use case element using vector space model. Recall, precision and harmonic mean are used to evaluate the proposed approach. Thitiwan Sriudorn [5] proposed approach to store and retrieve use case description for SPL using vector space model. Also use recall, precision and harmonic mean are used to evaluate the proposed approach. The both research are store and retrieve software requirements specification. Our research expands the scope of SPL modeling by including SRM and SAM based on UML that involved with SPL features. The differences between these three researches are showed in Table I.

TABLE I
THE DIFFERENCE BETWEEN RESEARCHERS BASED ON VECTOR SPACE MODEL

| Author | Type of Document use in Research | Document and Model Storage and Retrieval |
|---|---|---|
| Akadej | Software requirement | Use case description |
| Thitiwan | Software requirement,<br>Software product line | Use case description |
| This research | Software requirement model,<br>Software analysis model,<br>Software product line | Use case model, Feature model, Static model, Dynamic model |

## III. SOFTWARE PRODUCT LINE

Software product line (SPL) [1] consists of a group of small software components integrated together to serve the execution of the common, optional and alternative feature. The advantage of SPL is the reuse of the outputs/results of the previous project to create a new system that may have difference software system or similar software system. An approach to develop SPL is considering family of software system and analyzing what features of the software family are common, optional or alternative. Common feature is feature that all members of family must have, optional feature is feature that some members of family have and alternative feature is feature that different versions of which are required by different members of the family.

Software product line engineering phases are iterative software development process based on the driving of use case concept. Software product line engineering phases are depicted in Fig. 1

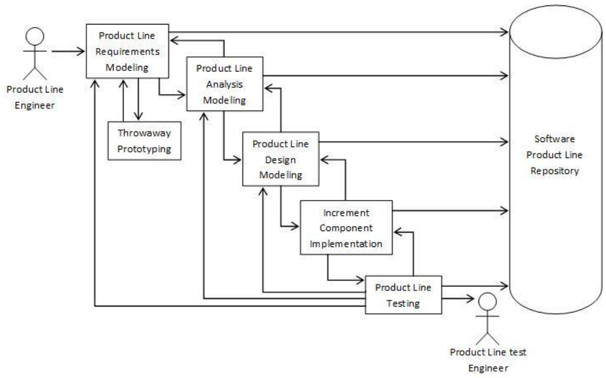

Product Line Engineer
Product Line Requirements Modeling
Product Line Analysis Modeling
Throwaway Prototyping
Product Line Design Modeling
Increment Component Implementation
Product Line Testing
Product Line test Engineer
Software Product Line Repository

Fig. 1 Software product line engineering phase [1]

During the product line requirements modeling phases, requirements models are developed. It consists of use case model and feature model.

During product line analysis modeling phases, static and dynamic models are developed.

During the product line design modeling phases, component-based software architecture of product line is developed.

During the increment component implementation phases, source code and increment component are developed.

After increment component implementation phases, the product line is tested by integration testing and functional testing of product line.

Modeling software product lines by integrating UML calls PLUS (Product Line UML-Based Software Engineering). PLUS use notation in UML to support the design of SPL process. This approach is similar to UML-based object-oriented method to model and analyze the system. It describes how to analyze the commonality and variability features of SPL.

In the software engineering process, it describes the relationship between software feature and use case due to the feature extraction from requirements. It can represent in a form of use case package or meta class to describe relation and types of feature. It also represent in a form of feature/use case dependencies table. An example of feature/use case dependencies table of microwave oven system is shown in Table II. It consists of feature name, feature category, use case name, use case category/variation point (VP) and variation point name.

TABLE II
AN EXAMPLE FEATURE/USE CASE DEPENDENCIES TABLE [1]

| Feature Name | Feature Category | Use Case Name | Use Case Category/Variation Point (VP) | Variation Point Name |
|---|---|---|---|---|
| Microwave Oven Kernel | common | Cook Food | kernel | - |
| Light | optional | Cook food | vp | Light |
| English | alternative | Cook Food | vp | Display Language |

Furthermore, it describes the relationship between features and class according to the functional requirements. The relation is represented by feature-based class diagram or feature/class dependencies table. This paper uses feature/class dependencies table to describe relation between feature and class. An example of feature/class dependencies table is shown in Table III. It consists of feature name, feature category, class name and class category.

TABLE III
AN EXAMPLE FEATURE/CLASS DEPENDENCIES TABLE [1]

| Feature Name | Feature Category | Class Name | Class Category |
|---|---|---|---|
| Microwave Oven Kernel | common | Door Sensor Interface | kernel |
| | | Weight Sensor Interface | kernel-abstract-vp |
| | | Keypad Interface | kernel-param-vp |

A SPL modeling with UML, use case are assigned with stereo type for classify type of use case which is kernel use case, optional use case or alternative use case. Kernel use case are required by all member of the product line, optional use case are required by some member of the product line and alternative use case has different version of the use case are required by different member of the product line. An example use case for SPL is shown in Fig. 2.

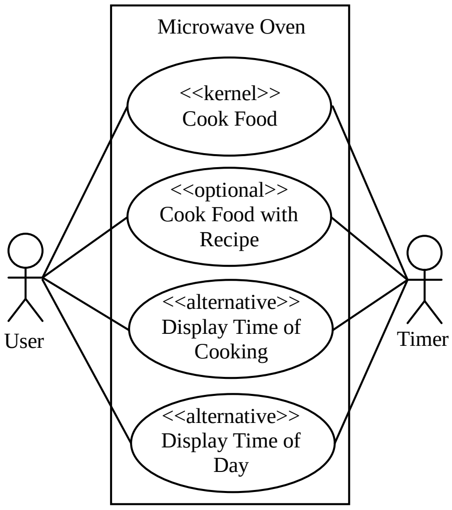


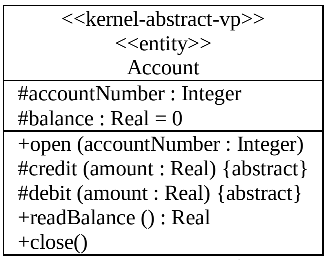
Fig. 2 An example use case for SPL [1]

Use case description for SPL is different from normal use case. It has additional part from normal use case are reuse category and variation point. Reuse category refers to types of use case which is kernel, optional or alternative use case. Variation point is sub use case. Each use case may have one or more variation point. It consists of variation point name, type of functionality, line numbers are shown in use case description and description of functional. An example use case description for SPL is shown in Table IV.

TABLE IV
AN EXAMPLE USE CASE DESCRIPTION FOR SPL [1]

| Software Product Line Use Case Description | |
|---|---|
| Use Case Name | |
| Reuse Category | |
| Summary | |
| Dependency | |
| Actor | |
| Precondition | |
| Description | |
| Alternative | |
| Postcondition | |
| Variation Point | |
| Name | |
| Type of Functionality | |
| Line Numbers | |
| Description of Functional | |

Class for SPL has a stereo type for classify type class by SPL. It is reused characteristics of the product line classes. In UML notation, a stereo type is enclosed by guillemets, like this: <<kernel>>. An example CRC Card for SPL is shown in Fig. 3.

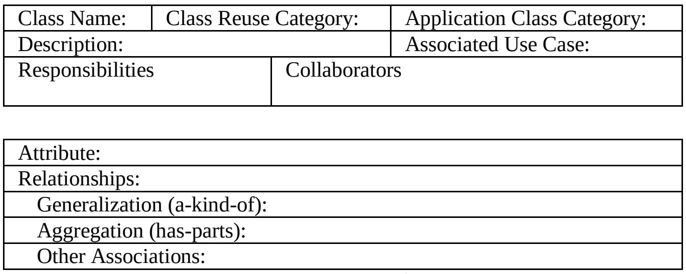

<<kernel-abstract-vp>>
<<entity>>
Account

#accountNumber : Integer
#balance : Real = 0

+open (accountNumber : Integer)
#credit (amount : Real) {abstract}
#debit (amount : Real) {abstract}
+readBalance () : Real
+close()

Fig. 3 An example class for SPL [1]

Class-Responsibility-Collaboration Card (CRC Card) for SPL is applied from normal CRC Card by addition class reuse category and application class category. An example CRC Card for SPL is shown in Fig. 4.



| | | |
|---|---|---|
| Class Name: | Class Reuse Category: | Application Class Category: |
| Description: | | Associated Use Case: |

| | |
|---|---|
| Responsibilities | Collaborators |

| |
|---|
| Attribute: |
| Relationships: |
| Generalization (a-kind-of): |
| Aggregation (has-parts): |
| Other Associations: |

Fig. 4 An example CRC Card for SPL [1], [11]

State chart diagram and sequence diagram for SPL are the same as normal UML diagram.

This paper proposes an approach to store and retrieve requirement and analysis model which consist of use case diagram, use case description, class diagram, CRC Card, state chart diagram, sequence diagram, feature/use case dependencies table and feature/class dependencies table. We conclude the difference between the UML for SPL and normal UML model are shown in Table V.

TABLE V
THE DIFFERENT BETWEEN NORMAL MODELING AND SPL MODELING

| Requirement Model and Analysis Model | Normal Modeling | SPL Modeling |
|---|---|---|
| Use Case Diagram | Based on UML | Add Stereo Type to Categorize Use Case |
| Use Case Description | Based on UML | Add Reuse Category and Variation Point |
| Class Diagram | Based on UML | Add Stereo Type to Categorize Class |
| CRC Card | Based on UML | Add Reuse Category and Application Class Category |
| State Chart Diagram | Based on UML | Based on UML |
| Sequence Diagram | Based on UML | Based on UML |
| Feature/Use Case Dependencies Table | Not Have | Have |
| Feature/Class Dependencies Table | Not Have | Have |

## IV. OUR APPROACH FOR SPL REQUIREMENTS AND ANALYSIS MODEL STORAGE AND RETRIEVAL

Information storage and retrieval [6], [7] are processes related to the store, manage, access and present information in an ease of use form to user. To retrieval information from the specified collection, user enter query according to his/her interest for the similarity computation

Storage and retrieval of requirement model and analysis model for SPL processes have two modules. They are the storage requirement model and analysis model for SPL module and the retrieval requirement model and analysis model for SPL module are shown in Fig. 5.

### A. *Importing model and document*

Models and documents are imported and converted to XML document for the relationship creation between models and the preparation for index term weighted value computation.

### B. *Indexing*

All words appear in all models and documents are indexed using inverted file format. The term frequency in each model or document and the total frequency in all are documented.

### C. *Computing index weighted value*

Index weighted value computed by (1).

$$W_{ik} = \frac{\mathrm{Freq}_{ik}}{\mathrm{TotFreq}_{k}} \qquad (1)$$

Where:
$W_{ik}$ = index weighted value k in document i
$\mathrm{Freq}_{ik}$ = all index frequency k in document i
$\mathrm{TotFreq}_{k}$ = all index k in all documents of database

Index weighted value is used to compute the similarity between term/key word appeared in document and user's query.

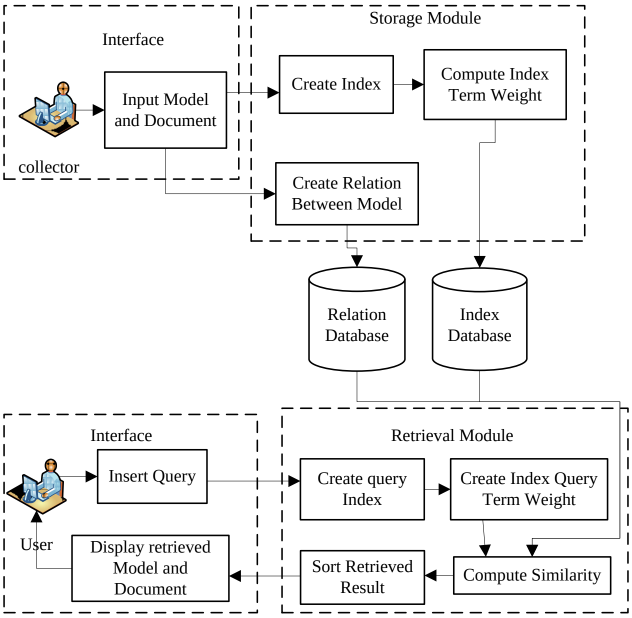


Fig. 5 Storage and retrieval requirement model and analysis model approach

### D. *Creating relation between models*

When all model and document are extracted in a form of text document or XML document, relationship between each model is created using entity relationship diagram as shown in Fig. 6. All model and document are stored in the database.

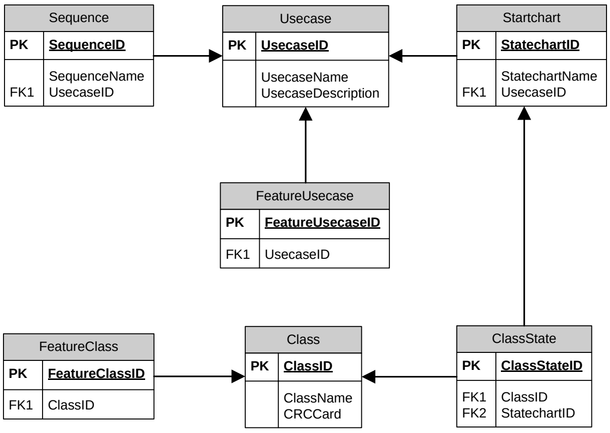


Fig. 6 An example of relationship between models

### E. *Inputting queries*

User enters query for the retrieval of user's interest document using key word and Boolean operators such as AND and OR.

### F. *Queries indexing*

After user enters query, each query is used to create query's index for query index weighted value computation.

### G. *Query index weighting*

Use (1) to compute query index weighted value to compute similarity score between query and key word appeared in collected documents.

### H. *Computing similarity*

Vector space cosine coefficient similarity computation is used to identify the similarity between query and key word in document collection as shown in (2).

$$\mathrm{SIM}(Q_i, \mathrm{DOC}_j) = \frac{\sum_{k=1}^{N}(W_{i,k} \cdot W_{j,k})}{\sqrt{\sum_{k=1}^{N}(W_{i,k})^2 \cdot \sum_{k=1}^{N}(W_{j,k})^2}} \qquad (2)$$

Where:
$\mathrm{SIM}(Q_i, \mathrm{DOC}_j)$ = similarity scores between query $Q_i$ and $\mathrm{DOC}_j$
$W_{i,k}$ = weighted value of term k in query i
$W_{j,k}$ = weighted value of term k in query j

### I. *Sorting retrieved result*

After computing the similarity for all models and documents, the results of computing similarity score are sorted and assembly for presenting all retrieved models and documents to user for ones above the selected threshold.

### J. *Displaying result*

The sorted results of retrieval according to similarity score are displayed to user.

In retrieval evaluation, this paper present three metrics to evaluate retrieval result relevant to user's query. Three metrics are recall, precision and harmonic mean [6], [7]. Recall is the proportion of relevant documents which has been retrieved and all relevant documents in the collection. Precision is the proportion of relevant documents which has been retrieved and retrieved document. Harmonic mean is a single measure which combines recall and precision.

## V. Case Study

This section presents a case study to illustrate our concept for assigning weight for index term of document and similarity computing between collected document and user's query.

Suppose the first system consists of requirement model and analysis model for SPL consists of eight types of document. They are use case diagram ($UC_1$), feature/use case dependencies table ($FUC_1$), use case description ($UCD_1$), class diagram ($C_1$), feature/class dependencies table ($FC_1$), CRC Card ($CRC_1$), sequence diagram ($SQ_1$) and state chart diagram ($ST_1$)

Suppose the second system consists of requirement model and analysis model for SPL consists of eight types of document as the first system ($UC_2$, $FUC_2$, $UCD_2$, $C_2$, $FC_{2,}$ $CRC_{2,}$ $SQ_2$ and $ST_2$)

Suppose frequency of index contained in requirement model and analysis model are $T_1$, $T_2$, $T_3$, $T_4$ and $T_5$ and are shown in Table VI. TotFreq is total frequency index of each index in requirement model and analysis model for SPL.

TABLE VI
THE FREQUENCY OF INDEX WHICH APPEARS IN DOCUMENT

| DOC | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $UC_1$ | 3 | 2 | 1 | 4 | 1 |
| $FUC_1$ | 10 | 1 | 0 | 3 | 2 |
| $UCD_1$ | 4 | 0 | 1 | 1 | 2 |
| $C_1$ | 1 | 1 | 0 | 0 | 1 |
| $FC_1$ | 3 | 0 | 2 | 2 | 3 |
| $CRC_1$ | 1 | 1 | 0 | 1 | 1 |
| $SQ_1$ | 0 | 1 | 0 | 1 | 0 |
| $ST_1$ | 0 | 0 | 0 | 0 | 0 |
| $UC_2$ | 1 | 7 | 0 | 3 | 1 |
| $FUC_2$ | 2 | 6 | 1 | 5 | 0 |
| $UCD_2$ | 0 | 12 | 1 | 4 | 1 |
| $C_2$ | 2 | 1 | 0 | 1 | 3 |
| $FC_2$ | 1 | 0 | 1 | 2 | 1 |
| $CRC_2$ | 0 | 2 | 0 | 1 | 0 |
| $SQ_2$ | 0 | 0 | 1 | 0 | 1 |
| $ST_2$ | 1 | 0 | 1 | 1 | 1 |
| TotFreq | 29 | 34 | 9 | 29 | 18 |

Index weighted value computed by (1). The results are shown in table VII.

TABLE VII
THE WEIGHTED VALUE OF EACH INDEX

| DOC | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $UC_1$ | 3/29 | 2/34 | 1/9 | 4/29 | 1/18 |
| $FUC_1$ | 10/29 | 1/34 | 0/9 | 3/29 | 2/18 |
| $UCD_1$ | 4/29 | 0/34 | 1/9 | 1/29 | 2/18 |
| $C_1$ | 1/29 | 1/34 | 0/9 | 0/29 | 1/18 |
| $FC_1$ | 3/29 | 0/34 | 2/9 | 2/29 | 3/18 |
| $CRC_1$ | 1/29 | 1/34 | 0/9 | 1/29 | 1/18 |
| $SQ_1$ | 0/29 | 1/34 | 0/9 | 1/29 | 0/18 |
| $ST_1$ | 0/29 | 0/34 | 0/9 | 0/29 | 0/18 |
| $UC_2$ | 1/29 | 7/34 | 0/9 | 3/29 | 1/18 |
| $FUC_2$ | 2/29 | 6/34 | 1/9 | 5/29 | 0/18 |
| $UCD_2$ | 0/29 | 12/34 | 1/9 | 4/29 | 1/18 |
| $C_2$ | 2/29 | 1/34 | 0/9 | 1/29 | 3/18 |
| $FC_2$ | 1/29 | 0/34 | 1/9 | 2/29 | 1/18 |
| $CRC_2$ | 0/29 | 2/34 | 0/9 | 1/29 | 0/18 |
| $SQ_2$ | 0/29 | 0/34 | 1/9 | 0/29 | 1/18 |
| $ST_2$ | 1/29 | 0/34 | 1/9 | 1/29 | 1/18 |

Suppose user's queries (Q) are $T_1$, $T_3$ and $T_5$. The similarity between collected document and user's query computed by (2) and the results are shown in table VIII.

TABLE VIII
THE SIMILARITY SCORE BETWEEN COLLECTED DOCUMENT AND QUERIES

| DOC | SIM Q($T_1$,$T_3$,$T_5$) | DOC | SIM Q($T_1$,$T_3$,$T_5$) |
|---|---|---|---|
| $UC_1$ | 0.707271 | $UC_2$ | 0.217043 |
| $FUC_1$ | 0.696552 | $FUC_2$ | 0.372333 |
| $UCD_1$ | 0.981245 | $UCD_2$ | 0.080433 |
| $C_1$ | 0.725041 | $C_2$ | 0.731493 |
| $FC_1$ | 0.934015 | $FC_2$ | 0.794293 |
| $CRC_1$ | 0.6534 | $CRC_2$ | - |
| $SQ_1$ | - | $SQ_2$ | 0.816497 |
| $ST_1$ | - | $ST_2$ | 0.870209 |

The results of the similarity computation between requirements model and analysis model in the collection and user's query are retrieved and sort as $UCD_1$, $FC_1$, $ST_2$, $SQ_2$, $FC_2$, $C_2$, $C_1$, $UC_1$, $FUC_1$, $CRC_1$, $FUC_2$ and $UC_2$.

From the calculated results are summarized as follows. $SQ_1$, $ST_1$ and $CRC_2$ cannot be retrieved because there is no index term that matches the user's query. For SPL development, user may require a model that involves the retrieval model. Table VIII shows the relation between models. When user wants other documents that related to the document being retrieved, it can bring more the related document to present to the user. In this case study, $UCD_1$ meets the requirements of user. Whenever user wants relevant document which is $SQ_1$, this approach can show $SQ_1$ to user because we create the relationship between documents that are presented in Fig. 6.

## VI. Conclusion and Future work

This paper presents the approach for the storage and retrieval of requirements model and analysis model for SPL with the application of the information storage and retrieval principle. The case study is also presented to show how to implementing our approach. We plan to develop a software tool to support our concept and analyze it based on recall, precision and harmonic mean evaluation criteria.

## References

[1] Hassan Gomaa, "Designing Software Product Line with UML", Addison-Wesley, 2004.

[2] Hussan Gomaa and Michael E. Shim, "Automated Software Product Line Engineering and Product Derivation", 40th Hawaii International Conference on System Sciences (HICSS'07), 2007.

[3] Paulo Gomes, Nuno Seco, Francisco C. Pereira, Paulo Paiva, Paulo Carreiro, José Ferreira, Carlos Bento, "An Approach to Software Design Reuse Using Case-Based Reasoning and WordNet", Proceeding of the 2006 conference on Integrated Intelligent Systems for Engineering Design, 2006.

[4] Akadej Udomchaiporn, Nakornthip Prompoon, and Pizzanu Kanongchaiyos, "Software Requirements Retrieval Using Use Case Terms and Structure Similarity Computation", XIII Asia Pacific Software Engineering Conference (APSEC'06), 2006.

[5] Thitiwan Sriudorn and Nakornthip Prompoon, "Use Case Description Storage and Retrieval for Software Product Line", NCSEC 2007.

[6] Ricardo Baeza-Yates and Berthier Ribiero-Neto, "Modern Information Retrieval", Addison Wesley, 1999.

[7] Gerard Salton, M.j. McGill, "Introduction to Modern Information Retrieval" McGraw-Hill. New York, 1983.

[8] Booch G, Rumbaugh J, and Jacobson I, "The Unified Modeling Language User Guide SECOND EDITION", Addison Wesley Professional, 2005.

[9] Larman C, “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”, Addison Wesley Professional, 2004.

[10] Elliotte Rusty Harold, “XML 1.1 Bible 3rd Edition”, Wiley Publishing Inc, 2004.

[11] Alan Dennis, Barbara Haley Wixom, David Tegarden, “Systems Analysis and Design with UML Version 2.0 AN OBJECT-ORIENTED APPROACH Second Edition”, John Wiley & Sons Inc, 2005.

[12] Hanh Nhi TRAN, Bernard COULETTE, Dan Thu TRAN, My Hang VU, “Automatic Reuse of Process Patterns in Process Modeling”, SAC’11, 2011.

[13] Markus Luckey, Andrea Baumann, Daniel Mendez Fernandez, Stefan Wagner, “Reusing Security Requirements Using an Extanded Quality Model”, SESS’10, 2010.

[14] F. Deissenboeck, S. Wagner, M. Pizka, S. Teuchert, and J. F. Girard, “An activity-based quality model for maintainability, In Proc, IEEE International Conference on Software Maintenance (ICSM 2007), 2007