Requirements Association Extraction based on Use Cases Diagram

Peter Gelu, Riyanarto Sarno, Daniel Siahaan

Department of Informatics, Institut Teknologi Sepuluh November
Surabaya, Indonesia

petergelu0803@gmail.com

Abstract

Requirements are the initial step in the software development process. It is very important to ensure the association (relationship) of requirements and high quality of specification as more than three-quarters failure of software derives from the software requirement process. Therefore, the analysis process is required to ensure the association between requirements and the requirements of other users. As a result, interdependency requirements association is essential. This research proposes an approach of software requirements association. These are based on the reference of interdependency in the user case, namely the result of collaboration of the association of the dependency of requirements based on Unified Modeling Language (UML) design in the Use cases diagram. In this research, the mapping between requirements and use cases and interdependencies between use cases are used to determine the interdependency between requirements. The analysis presented is the association of similar requirements, requires, or, temporal, elaborates dan generalises. The purpose of this research is to generate a requirements dependency graph that models the type of dependency between requirements within a software project.

Keywords: *Requirements, Use cases, Dependency Graph, Unified Modeling Language.*

1. Introduction

Requirements are an early stage in the process of developing software. Ensuring the association of requirements with the high quality of specification is essential, as more than three-quarters of system failures are derived from the of requirements analysis process [1]. The visualization of standard requirements and management of software engineering process is a report that identifies product or operational process, function, design characteristic or certain constraint and measurable [2]. The reseach is undertaken to obtain the results of the stages developed by software developer to enhance non-functional quality in the process of software upgrading [3]. According to Siahaan [4], there are two types of requirements, volatile requirements and persistent requirements. The persistent requirements are those which constant overtime, while volatile requirements are changed over time. Changes in a specific requirement statement can also occur at any time throughout the software development process. Thus this not only has a vertical impact, i.e. changes in designs and implementation, but also horizontal impact, changes in other depending requirements. Project manager who does not consider the horizontal impact may create a project cost estimation that has a higher mean error compare to its actual cost. This lead to a condition where project is over budget.

Goknil defines the types of interdependence association between requirements and reasoning about the dependency association of requirements by using [5] formalization to review the consistency of association requirements and concludes the association of new requirements. In the other research about engineering requirements proposed an advanced methodology for requirements solution technical engineering [6]. During the requirements process, the technics used are ordering requirements based on the priority and releasing the implementation. These are the essential steps undertaken in taking important decision to identify and analyze the priority technics in the requirements context [7]. The development of software requirements towards clients expectation is important. As a result, it is required to have the expert technician in software development. Each organization expects qualified and reliable software technology to perform well. Building a model approach of interdependency requirements and use cases are the result of relation interdependency collaboration based on Unified Modeling Language (UML), namely use cases diagram.

This research proposed a method to build the dependency graph by utilizing associatios between requirements and use case in the use case diagram. The stages in the extraction requirements process covered are metadata requirement extraction with the use case, association mapping requirements to use cases, and development of dependency graph association among requirements. The next extraction stages are extraction in the name and description of the use case.

2. The Use cases and Requirements

Use cases diagram is used for describing interaction between the user of the system (actor) with the use cases according to a certain scenario [8]. Since 1992, Jacob et al., used use case as a main model or requirement model in UML (*unified modeling language).* The use cases diagram can be seen in Figure 1.

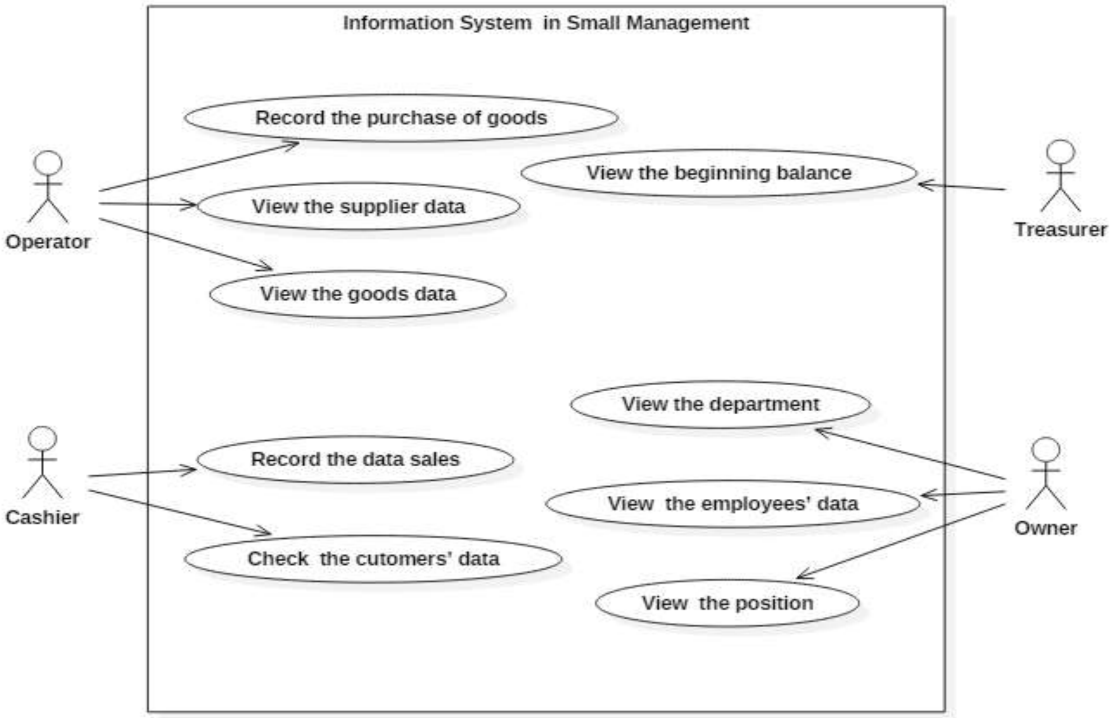


Figure 1. Use Cases Diagram of Information System in Small Management System

The use cases only describe what an actor perceives when interacting with the system. Therefore, the use cases can be linked to other use cases through some interdependency relationships: include, extend, alternative or specialization, and exception.

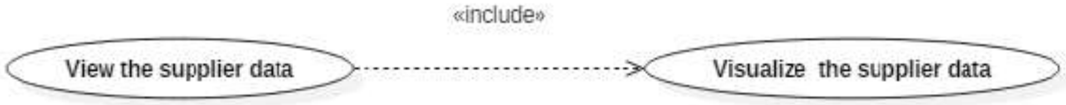


Figure 2. Include Interdependency Relationship

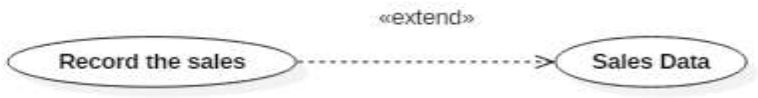


Figure 3. Extend Interdependency Relationship

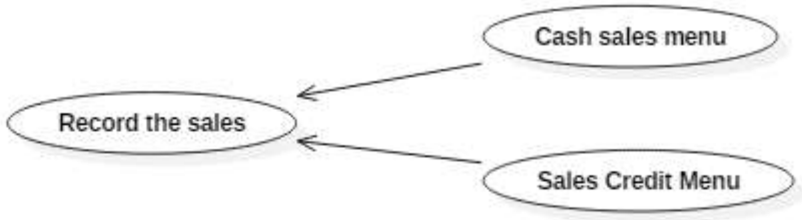


Figure 4. Alternative or Specialization Interdependency Relationship

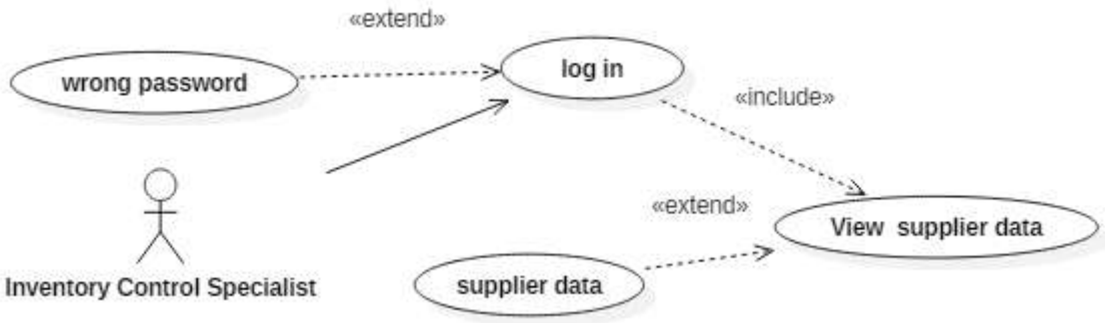


Figure 5. Exception Interdependency Relationship

The requirements engineering is a process of realizing a set of services that customers need for a software and the constraints when it is built or operated. Dahlstedt defines some of the interdepency requirements [9][10]. This research obtains some interdependency requirements, such as *Similar, Requires, Or, Temporal, Elaborates,* and *Generalises.* Table 1 shows these interdependencies.

Table 1. Mapping Relationship of Interdependency Requirements and Use Cases

No	Mapping	
	Requirement Dependency	**Use cases Relationship**
1	Similar	Alternative
2	Requires	Include
3	Or	Alternative
4	Temporal	Precondition
5	Elaborates	Extend
6	Generalises	Exception

3. Reseach Methodology

This research defines some processes to know the association among requirements, namely the association of Metada Requirements Extraction and Use cases. These processes extract every requirements and every use cases and description in the use cases. The extraction results of requirements and use cases in the forms of triplets to be used in the process of similarity approaches between requirements and use cases.

Processed data in this research is the process of extracting all requirements and use cases. The results of the process are extracted to get the value of similarity between requirements. On the other hand, actor, use cases name, and use cases description are taken based on the use cases diagram. Each requirement produces triplets that will measure the value of similarity in use cases. The results of this process will show the interdependency association between requirements and the use cases. This research proposes the flow of development requirements association based on use cases diagram. This can be seen in Figure 6.

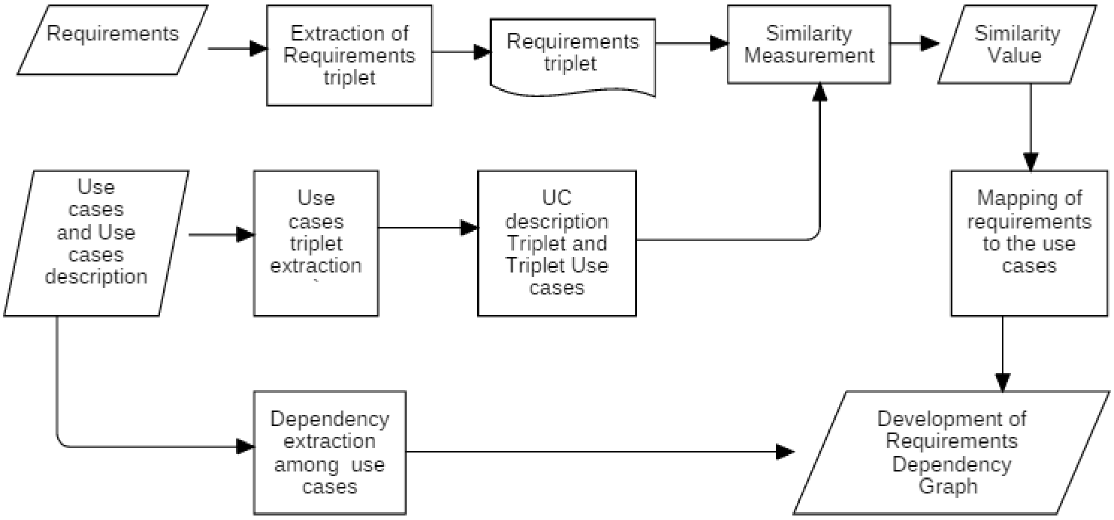

Requirements
Extraction of Requirements triplet
Requirements triplet
Similarity Measurement
Similarity Value
Use cases and Use cases description
Use cases triplet extraction
UC description Triplet and Triplet Use cases
Mapping of requirements to the use cases
Dependency extraction among use cases
Development of Requirements Dependency Graph

Figure 6. The Development Flow of Requirements Relationship Based on Use Cases Diagram

Table 2 shows the definition of functional and non-functional requirements of information systemin small business management.

Table 2. Specification of Functional Requirements of Information System in Small Business Management

ID	Statement
R1	Inventory officers can see the list of suppliers
R2	Inventory officers can record item purchasing data
R3	Cashier can record sales data
R4	The system provides a feature to fill multiple data at once
R5	Inventory officers must have access rights

The next process is extraction of requirements triplet. The process extracts all requirements in the text form into triplets of requirement. Figure 7 is a flow process of requiremenst extraction.

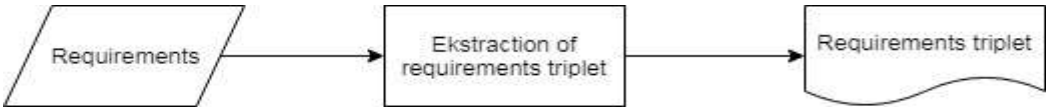


Figure 7. The Flow of Triplet Requirements Extraction

The extracted requirement triplet process consists of (Actor/System, Action, Object). To extract the requirements triplet, the statement document is taken on every line of the document. The first step to extract requirements is changing text document into the form of a sentence description. The next step is converting all text documents into a series of requirements triplet.

The results part-of speech tagging in the form of verbs and nouns, then checking the word through Wordnet needs to be done as well as for checking words that have no meaning. Furthermore, the tokenization stage so that the set of pre-processed verbs and nouns produces triplets as shown in Table 3.

Table 3. Triplet Requirement System

R1	{Inventory;officers},{can;see},{suppliers}
R2	{Inventory;officers},{can;record;purchasing},{item;data}
R3	{Cashier},{can;record},{sales;data}
R4	{System},{provides;fill},{feature;data}

This research proposes a case study of information system in small business management which focuses on the process of software requirement and each stage of the process of developing software. The following process is extracting two parts of use cases diagram which consists of the name of the use cases and the description.The process flow can be seen in the Figure 8.

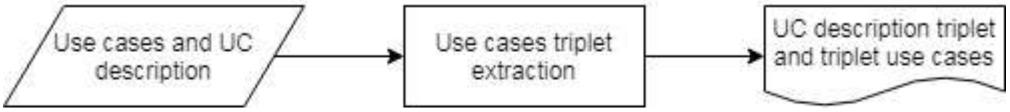
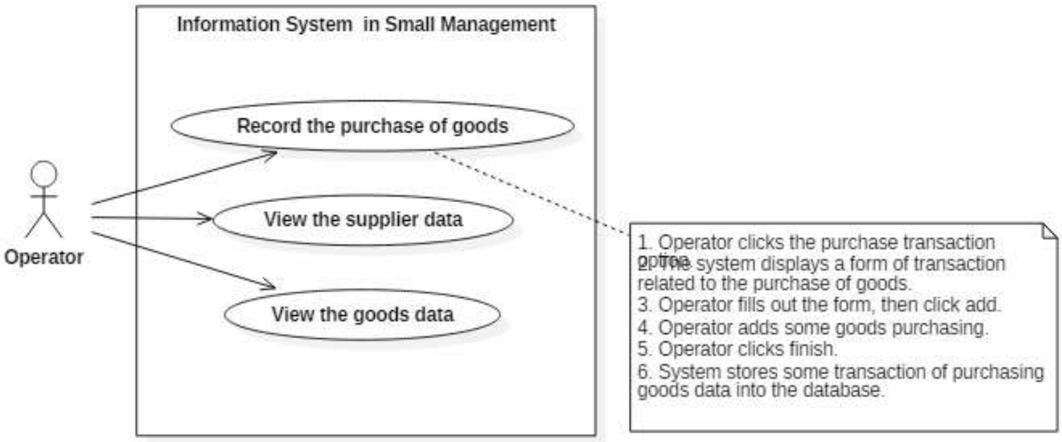

Use cases and UC description
Use cases triplet extraction
UC description triplet and triplet use cases

Figure 8. The Flow of Extraction Process of Use Cases Triplet and Use Cases Description

For the second triplet extraction, use cases descripton is extracted from the same diagram. Figure 9 displays the example of use cases and use cases description “record the purchase of goods” by separating into a form of sentence.


Information System in Small Management
Record the purchase of goods
View the supplier data
View the goods data
Operator

1. Operator clicks the purchase transaction option
2. The system displays a form of transaction related to the purchase of goods.
3. Operator fills out the form, then click add.
4. Operator adds some goods purchasing.
5. Operator clicks finish.
6. System stores some transaction of purchasing goods data into the database.

Figure 9. The Use Cases and Use Cases Description of Recording The Purchase of Goods

Basic flow of Use cases in Figure 9 is undertaken based on the sequence numbers with the result as follows :

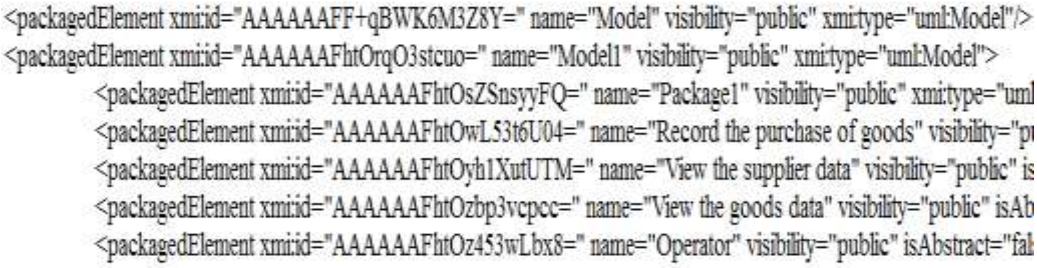
1. Operator clicks the purchase transaction option.
2. The system displays a form of transaction related to the purchase of goods.
3. Operator fills out the form, then click add.
4. Operator adds some goods purchasing.
5. Operator clicks finish.
6. System stores some transaction of purchasing goods data into the database.

The following stage of Natural Language Process based on the sequence numbers is converted into triplet through tokenaziton stage based on the use cases diagram as seen as the following result :

(operator;record;transaction;purchasing)
(system,view. Form, transaction, purchasing)

(operator;record; fill in ;form)
(operator;record;add; purchasing;goods))
(operator;record;finish)
(system, save;data;transaction;purchasing;goods;in;basis;data)

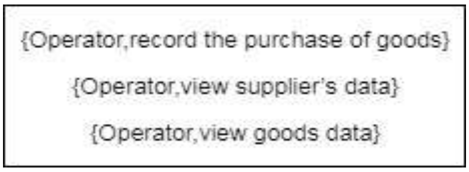
The Use cases triplet extraction process in the use cases are from the name of use cases and its description. The use cases diagram is exported into the XMI file. The result of XMI file can be seen in Figure 10.



```
<packagedElement xmi:id="AAAAAAFF+qBWK6M3Z8Y=" name="Model" visibility="public" xmi:type="uml:Model"/>
<packagedElement xmi:id="AAAAAAFhtOrqO3stcuo=" name="Model1" visibility="public" xmi:type="uml:Model">
        <packagedElement xmi:id="AAAAAAFhtOsZSnsyyFQ=" name="Package1" visibility="public" xmi:type="uml
        <packagedElement xmi:id="AAAAAAFhtOwL53t6U04=" name="Record the purchase of goods" visibility="pu
        <packagedElement xmi:id="AAAAAAFhtOyh1XutUTM=" name="View the supplier data" visibility="public" is
        <packagedElement xmi:id="AAAAAAFhtOzbp3vcpcc=" name="View the goods data" visibility="public" isAb
        <packagedElement xmi:id="AAAAAAFhtOz453wLbx8=" name="Operator" visibility="public" isAbstract="fal
```

Figure 10. XMI file extraction from use cases and use cases description.

Moreover, the XMI File generated in Figure 10 is processed through natural language processing stages namely part-of speech tagging and tokenization to extract actor pairs and use cases as in Figure 11.



{Operator,record the purchase of goods}
{Operator,view supplier's data}
{Operator,view goods data}

Figure 11. Result Extraction of Use Cases and Use Cases Description

The similarity measurement process is calculation of the value of similarity between the description of the use case and the requirement description. Calculations on triplet words based on subject, predicate and object via POS tagging stages. The description of the triplet requirement in Table 3 and the triplet description of the use case in Table 4 is made in the process of calculating the similarity values.

Table 4. Triplet Uses Cases and Use Cases Description

UC01	{Record},{purchase},{item}
UC02	{View},{supplier},{data}
UC03	{record},{sales},{data}

After a triplet process of words, searched for similarity values between triplet description of use cases and triplet descriptions of requirement is to look at the similarity association between words using WS4J online. Table 5 is a triplet calculation of functional requirement of R2 with triplet description of UC01 use case. The result of the calculation of similarity values in the form of matrix in Table 5 to be made in the process of building dependency graph requirement based on the association on the use case diagram.

Table 5. Triplet Similarity Value Calculation Use Case and Triplet Requirement (Wu-Palmer Calculation)

		UC01		
		record	purchase	item
R2	Inventory	0.8889	0.6316	0.6667
	officers	0.4348	0.375	0.5882
	Can	0.6667	0.4	0.6667
	record	1	0.6667	0.6154
	Item	0.6154	0.5714	1
	purchasing	0.6316	0.9474	0.4444
	Data	0.5	0.4286	0.5

Similarity value from Table 6 will be obtained from the result of triplet requirements process. The result of the process of calculating similarity value between requirements the example R1, R2, R4, R5 and use cases will generate a mapping process. This mapping will be used by developers to know the association among the requirements in the software development process. Moreover, the dependency graph among requirements (R1, R2, R4 and R5) also important to be obtained as shown in Figure 12.

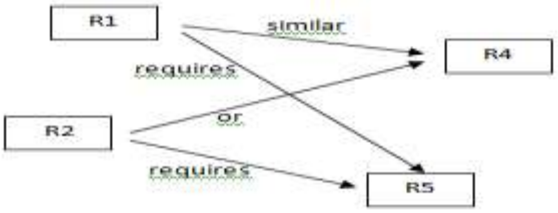


Figure 12. Development of Requirements Dependency Graph

The process of dependency association of *similar* defines the similarity association between the requirement of R1 and R4 when the requirement for R1 is changed will have an effect on the requirement of R4 and vice versa. Therefore, the requirements change process will also affect the software development process. The analysis from developer is important in order to obtain the requirements change process.

4. Results And Discussion

This research based on case study on information system in small business management (Simancil). This software is developed to support SME (Small & Medium Enterprises) in managing administration and finance to ease the business actors in arranging their business activity using computerized system. Some of the most beneficial features of SiMancil for SME management are Bookkeeping / Accounting, Sales / Purchasing, Product Stocks, Personnel, and Reports. Figure 1 is use cases diagram of information system in small management system. On the diagram there are 4 actors. The treasurer actor has the role to manage the finances of SME. This actor has the basic requirements to record the beginning balance, including cash out and cash in. Inventory Actor has the role to manages the stock of goods. This actor has the basic requirements to add the data suppliers, add customer data, and record purchase of goods from suppliers. The cashier actor has a role to be a contact person for the customer. The basic requirements of this actor is to add customer data and record the sales. Lastly, the owner actor who has the role of managing employee data in his business place as shown in Figure 1. Based on the use cases diagram in Figure 1, the requirement specification was obtained which consist of functional and non-functional requirements.

The process used to measure the similarity of software requirements to the use cases diagrams is by calculating the value of similarity between triplet extraction requirements and triplet extraction of use cases to obtain similarity value for subsequent processes in software development. The process of calculating the value of similarity between the use cases and the requirements description is on the triplet of words based on subject, predicate and object. After all descriptions of triplet use cases and triplet requirements are obtained, the similarity values were calculated between triplet descriptions of use cases and triplet descriptions of requirements by looking at similarity association between words using online WS4J (Wu-Palmer calculations) [11]. Moreover, Greedy's algorithm [12] is used to calculate the value of similarity between requirements and use cases as seen in Table 6.

Table 6. The Result of Calculating The Similarity Value Between Requirements and Use Cases

	R1	R2	R3	R4	R5
UC01	**0.54**	**0.58**	0.34	0.23	0.43
UC02	0.3	0.4	0.1	**0.55**	0.4
UC03	0.4	0.3	0.03	0.2	**0.65**

The result of the similarity value calculation shows the interdependency association between requirements and use cases. The result of the similarity value interdependency approach is shown in Figure 14.

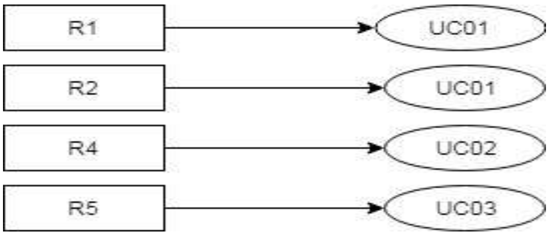


Figure 14. The Result Mapping Relationship of Requirements to The Use Cases

Figure 14 is used by the developer in the process of software development. It shows that there is an interdependency association between software requirements and use cases. The process described is as in the mapping of the interdependency association among the requirements toward use cases. This approach is undertaken on the system development process or the change in software requirements process.

5. Conclusion

This study proposes a new approach model for analyzing requirements interdependency based on document specifications. The methods presented in this research are the extraction of metadata requirements with use cases, mapping association between requirements with use cases, and the development of dependency graph association among the requirements. As a result, it forms association among requirements based on the use cases diagram.

The process of mapping the interdependency association between the requirements with the use cases has not been maximized in this study. There should be a re-analysis of the interdependency association between requirements in terms of value related, abstraction, content, condition and evolutionary. This is because the process of dependency association analysis only use a small part of some dependency association among requirements. Mapping the association between requirements and use cases in this research can be undertaken by approaching other artifacts in UML such as data flow diagrams, class diagrams, sequence diagrams, interaction diagrams, state diagrams and object diagrams. As a result, it produces a complex approach in the process of changing requirements in the software development. As a result, the requirements dependency graph can also be used to predict which sections or modules can be developed along side the change of requirements.

There is a need to evaluate the result of mapping the use case to requirements. This is to ensure that correct realization association is produced. Further research would be expanding the dataset with various project domains and different dependencies and artifacts, Thereby generating association between requiretment based on use case.

References

[1] M. M. Geogy, A. Dharani, and # Professor, “ScienceDirect A Scrutiny of the Software Requirement Engineering process,” *Procedia Technol.*, 2015.

[2] Z. S. H. Abad, M. Noaeen, and G. Ruhe, “Requirements Engineering Visualization: A Systematic Literature Review,” in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 2016.

[3] L. Globa, T. Kot, A. Reverchuk, and A. Schill, “Method of non-functional requirements balancing when service development,” *J. Theor. Appl. Comput. Sci.*, vol. 6, no. 3, pp. 50–57, 2012.

[4] D. Siahaan, “Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak,” *Yogyakarta Andi*, 2012.

[5] C. Arora, M. Sabetzadeh, A. Goknil, L. C. Briand, and F. Zimmer, “Change impact analysis for Natural Language requirements: An NLP approach,” in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, 2015.

[6] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. R. Mahrin, “A systematic literature review of software requirements prioritization research,” *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014.

[7] M. Batra and B. Archana, “Descriptive Literature Review of Requirements Engineering Models,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 2, pp. 289–293, 2015.

[8] S. Sabharwal, R. Sibal, and P. Kaur, “Deriving Complexity Metric Based on Use Case Diagram and its Validation,” in *2014 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2014*, 2014, pp. 102–107.

[9] Å. G. Dahlstedt, “Requirements Interdependencies – a Research Framework,” no. July, 2001.

[10] R. E. Bloomfield, P. Popov, K. Salako, V. Stankovic, and D. Wright, “Preliminary interdependency analysis: An approach to support critical-infrastructure risk-assessment,” *Reliab. Eng. Syst. Saf.*, vol. 167, pp. 198–217, 2017.

[11] P. Sharma, R. Tripathi, V. K. Singh, and R. C. Tripathi, “Automated patents search through semantic similarity,” *IEEE Int. Conf. Comput. Commun. Control. IC4 2015*, 2016.

[12] M. A.-R. Al-Khiaty and M. Ahmed, “Similarity Assessment of UML Class Diagrams using a Greedy Algorithm,” in *2014 International Computer Science and Engineering Conference (ICSEC)*, 2014, pp. 228–233.