FRAMEWORK FOR FAST BUILDING SOFTWARE REQUIREMENTS SPECIFICATION

Ponglikit Singchai and Wanchai Rivepiboon

Software Engineering Laboratory, Department of Computer Engineering, Faculty of Engineering
Chulalongkorn University, Thailand.

E-mail: ponglikit.s@student.chula.ac.th

Abstract

Software Requirements is an important process in software project. If the information is inaccurate or delayed, the next steps may have problems. If system analyst has the initial or suggestion documents which use software requirements specification of past software projects with similar knowledge, it can help system analyst to building the software requirements specification involves gathering requirements from users more efficient and faster. Therefore, we propose a framework for fast building the relationships of the software requirements specification of past software projects from the similarity of the requirements in each project using Bayesian network model for retrieving requirements with similar knowledge to the new project.

Keywords: *Bayesian Network, Sentence Similarity, Software Requirements Specification, SRS, User Feedback.*

Abstrak

Software requirements adalah proses penting dalam proyek perangkat lunak. Jika informasi ini tidak akurat atau tertunda, langkah berikutnya kemungkinan akan bermasalah. Jika analis sistem memiliki dokumen awal yang menggunakan *software requirement specification* dari proyek perangkat lunak di masa lalu dengan *knowledge* yang sama, maka hal tersebut dapat membantu analis sistem untuk membangun *software requirements specification* dengan lebih cepat dan efisien. Oleh karena itu, kami mem-*propose* kerangka kerja untuk membangun hubungan dari *software requirements specification* dari proyek perangkat lunak terakhir dengan melihat persamaan *requirements* dalam setiap proyek menggunakan model *Bayesian Network* untuk me-*retrieve requirements* dengan *knowledge* yang memiliki kemiripan pada proyek yang baru.

Kata kunci: *Bayesian Network, Persamaan Kalimat, Software Requirements Specification, SRS, Umpan Balik Pengguna.*

1. Introduction

The requirements process in software engineering is one of the important steps at the start of a project. If the information is inaccurate or delayed, the next steps may have problems [1].

At present, many software projects suffer from problems in which the project cannot successfully develop software effectively throughout the project. Some cases have been cause from incomplete or incorrect analysis of software requirements for users or business, which has been a major problem for software projects for a long time [2].

Software requirements analysis is a part of the planning stage of software project, which transforms user requirements to software requirements, which is the software requirements specification. However, system analysts cannot gather requirements from users correctly and completely, due to multiple factors such as: the users cannot explain all processes completely, the system analyst has little experience, etc. Also, software requirements specification is urgent for software development projects. If the software requirements specification is created faster than before, the new software project will be able to be started quickly. Therefore, the quick creation of a nearly complete software requirements specification is one of the important issues in software projects.

Therefore, we have a concept for applying the information of past successful software projects to be useful in the present [3] transformed from user requirements, consisting of functional requirements and non-functional requirements, then building relationships to link the relevant network using the theory of Bayesian network, to help in quickly obtaining a new software requirements specification which is nearly complete.

User requirements are retrieved from the relationship network for obtaining software requirements specification which is query or details for gathering and analyzing software requirements more complete and faster than before.

2. Background

2.1. Software Requirements Specification

The software requirements specification [4] is a document which is used in software development as an agreement of understanding between the user and the software developer. This research will store past software requirements specifications. In the past, according to the standard IEEE 830 [5] in a database for querying requirements, to help system analysts get software requirements specification that is very similar to the user requirement.

2.2. Sentence Similarity

Measuring sentence similarity [6] is to compare the similarity between two sentences, each of which is represented by a vector of size N (N-Dimensional Vector) to maintain the weight of each word in the sentence (N-Dimensional Vector in Term Space) to obtain the word similarity between two sentences, with which the sentence similarity with others sentences in software requirements specification is measured to determine whether each sentence is similar or not, for further determining the statistical probability.

Sentence similarity will be measured from the cosine of the angle between the two sentence vectors. If two sentences are very similar, the vectors of two sentences will overlap almost completely so the angle is very small, and the cosine value will be very high. The maximum value of the similarity measure in this method is equal to 1, which by definition is when the angle between the two vectors is 0 degrees, that is, the two vectors have the same direction. But the minimum value of the similarity measure in this way is equal to 0, which by definition is when the angle between the two vectors is 90 degrees, that is, the two vectors have different directions. The formula for sentence similarity is [7] as follows:

$$\boldsymbol{Sim(Si, Sj)} = \sum_{\boldsymbol{ti \,\epsilon\, Si}} \sum_{\boldsymbol{tj \,\epsilon\, Sj}} \frac{\boldsymbol{sim(ti,tj)}}{\sqrt{\boldsymbol{|Si||Sj|}}} \qquad (1)$$

From (1), if we measure the sentence similarity of 2 sentences, for which the first sentence is from the software requirements specification document and the second sentence is a software requirements specification consisting of purpose, scope, requirements, functional requirements, non-functional requirements, the similarity value is 0.47.

2.3. Bayesian Network

A Bayesian Network [**8**], [**9**] is a probability relationship which is cause-and-effect related. In this research, the various parts of software requirements specification are linked for with a relationship of the probability that if the event A occurs, then the event B will occur too.

We create the network by representing the nodes with topics in software requirement specification, namely, purpose, scope, user requirements, function requirements, non-functional requirements instead to link with probability in same project, which uses a probability formula for finding the relationships of software requirements specification as follows:

$$P{:}\,F, NF, R, S, P) = P(F|R,S,P) * P(NF|R,S,P) * P(R|S,P) * P(S|P) \quad (2)$$

Equation (2) is used for creating initial software requirements specification based on the probability of Bayesian network.

3. Related Works

In 2010, Anas Mahmoud proposed software requirements retrieval from functional requirements of past projects by storing requirements from successful projects as a library of functional requirements, which is a database [10] that a system analyst can input a query to retrieve functional requirements as shown in Fig. 1.

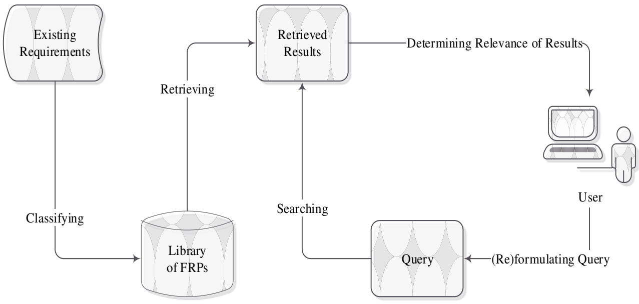

Fig 1. Storage and retrieval in functional requirements.

From Fig 1, when user inputs a query, the query will retrieve functional requirements in the database for obtaining the retrieved results for user to decide.

From this research, we have a concept of using information other than functional requirements, such as the purpose of the project, the scope of the project, software requirements, and non-functional requirements to create a relationship of each part for compositing the initial software requirements specification, which will allow gathering requirements process to be achieved more quickly.

In 2010, Tao Zhang proposed software requirements retrieval using semantic ontology and user feedback, which is retrieved in many domains, and using the user feedback for adjusting the weight of query [11], for obtaining more precise information as shown in Fig. 2.

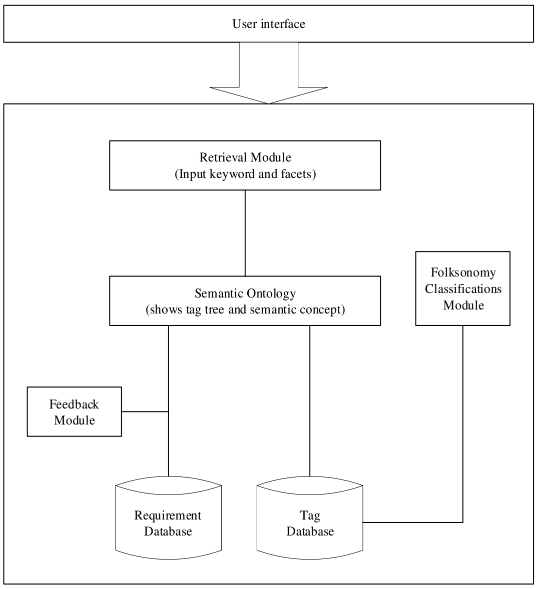
User interface
Retrieval Module
(Input keyword and facets)
Semantic Ontology
(shows tag tree and semantic concept)
Folksonomy Classifications Module
Feedback Module
Requirement Database
Tag Database

Fig 2. Retrieval by semantic ontology and user feedback.

From Fig 2, when the user inputs a query or factor, then semantic ontology will identify specific requirements or the relevant domain. After the system shows the results, then the user can give a score on how correct they think that these requirements are.

From this research, we have a concept of using user feedback to assist in more accurate retrieval of requirements.

4. Methodology

From the new concepts for quickly building software requirements specification, the steps can be described as detailed as the following:

2.1. Separating the structure of the software requirements specification

We separate the software requirements specification into the following topics in software requirements specification using the standard IEEE830:

1. Purpose of Project
2. Scope of Project
3. Software Requirements
4. Functional Requirements
5. Non-functional Requirements

The relationship of each topic is shown in Fig. 3.

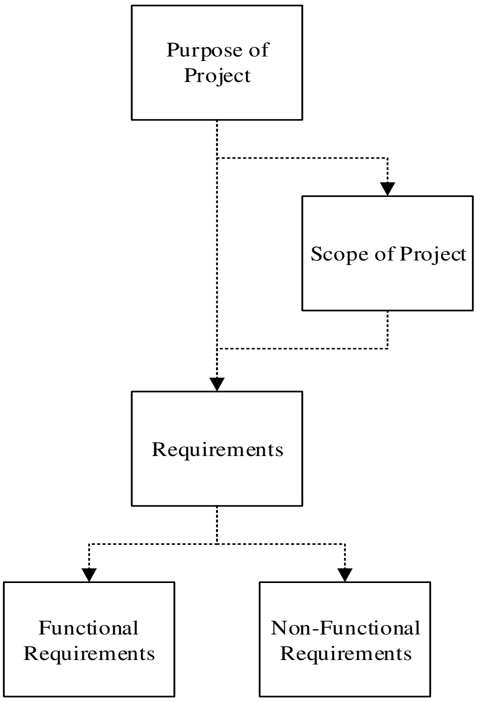

Fig 3. Relationship of topics in software requirements specification

The content in Fig. 3 shows the relationship of topics in software requirements specification. When starting a software project, the purpose of the project is created for explaining the purpose of the project clearly and understand what is doing. After obtaining the purpose of the project, then the scope of the project needs to be identified. The project and scope of project is relationship in same direction. And scope of project extend clearly for purpose of project. When the purpose and scope of the project are obtained, then gathering and analyzing the user or business requirements which same direction as purpose and scope of project for knowing the true requirements and something which develop initial. When user requirements are obtained, then the analysis and design are performed for creating functional requirements for developing what function the software should have and understand in more detail. Also, the user requirements can be analyzed for system performance requirements or non-functional requirements which it cannot identify functional requirements clearly, such as, system must complete the work in ten second.

When separating sentences in each topic in the software requirements specification, it is checked whether each sentence in same topic is similar or not, by measuring sentence similarity.

2.2. Measure sentence similarity

After separating sentences in each topic in the software requirements specification, the sentence similarity is measured in each sentence within the same topic in software requirements specification from the previous step using (1), which will produce a similarity value between 0 and 1 to determine whether or not, the sentences are the same. If similarity value is less than 0.9, they are different sentences. If similarity value is more than 0.9, they are the same sentences.

Afterwards, statistics for probability are created from past software projects using the theory of Bayesian network to create a database of relationship of software requirements specification of past software projects.

2.3. Building Bayesian Network using probability of relationship on past software projects

After measuring sentence similarity in each sentence, statistics from information of past software projects are obtained using same sentence in same topic, then related sentences are retrieved from subsequent topics such as purpose and scope A occur requirement B, C, D how much probability. If the sentences for the purpose and scope differ from purpose and scope A, it will be purpose and scope B. The probability of relationship is shown in Fig.4, which is a database of relationship of software requirements specification of past software projects for retrieval for create initial software requirements specification as shown in Fig. 4.

The content in Fig. 4 shows an example of probability of relationship using the theory of Bayesian network, between the purpose and scope of the software project and the user requirements, showing the statistics occurring in past software projects between the purpose and scope of software project and the user requirements. We assume that the probability of purpose and scope of software project A and user requirements C equals 0.6. The probability of purpose and scope of software project A and user requirements D equals 0.7. The probability of purpose and scope of software project A and user requirements E equals 0.9. The probability of purpose and scope of software project B and user requirements F equals 0.8. The probability of purpose and scope of software project B and user requirements G equals 0.2.

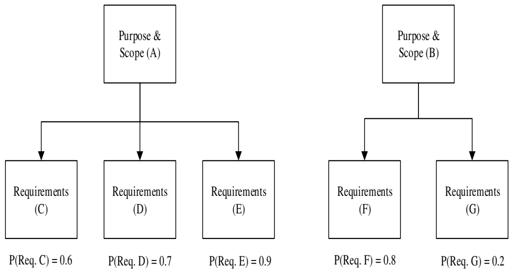

Fig 4. Example of probability of relationship using Bayesian network theory

2.4. Retrieval for building the initial software requirements specification

After storing the database of relationships of software requirements specification of past software projects, when the system analysts gathers user requirements, they will input the user requirements into the system in a query pattern, which the system will compare between the query and purpose and scope of past software projects that are most similar, using sentence similarity measurement in (1). After obtaining the purpose and scope of similar software projects, and obtaining relationships linked with purpose and scope of project, user requirements, functional requirements, and non-functional requirements with good probability are then obtained. They are shown to be initial software requirements specification which can be analyzed by the user and can be summarized quickly and nearly completely.

2.5. User Feedback

After system analysts retrieve and analyze the initial software requirements specification with the user to obtain the complete software requirements specification, system analysts will give a score on whether the retrieved results are correct or not, by giving a score between 1 and 5, ranging from least to most, for calculating the weight of correctness for that initial software requirements specification, which is information for adjusting the weight and obtaining better initial software requirements specification which better meets user requirements.

5. Results

This framework is used to store and retrieve information, using sentence similarity measurement and Bayesian network, creating a new framework for quickly building a software requirements specification.

From using the Bayesian network, we know the structure of past related software requirements specifications for creating new software requirements specification, by using the principles of Bayesian network which is about probability to assist in predicting new projects that use past software requirements specification is.

In addition, identifying which statistics that should need to know that is same or not so measuring sentence similarity to help determine whether it is the same or not.

6. Conclusions

This framework is a conceptual method that will help to create software requirements specification easily and quickly. Software requirements specifications of past software projects are used for building the relationship of each part, by using the theory of the Bayesian network, which predicts more accurately. New software requirements specification is like the initial or suggestion documents which help system analysts to create software requirements specification quickly. But it also requires systems analyst and users to think carefully to obtain software requirements which are correct and complete. In addition, it also relies on information of the software requirements specifications of other successful software projects.

7. Future Works

In the future, we plan to transform this framework into a tool which helps gather and analyze the initial software requirements quickly and correctly. Also, we will test the new tool with a software development team for developing similar software further.

References

[1] Joachim Karlsson, “Software Requirements Prioritizing”, in IEEE Int. Conf. on Requirements Engineering - RE, pp. 110-116, 1996.

[2] Hajar Mat Jani, Jalan Kajang-Puchong, Selangor Darul Ehsan, “Applying Case-Based Reasoning to software requirements specifications quality analysis system”, 2010.

[3] Wayne C. Lim, “Effects of Reuse on Quality, Productivity, and Economics”, in IEEE Software - SOFTWARE, vol. 11, no. 5, pp. 23-30, 1994.

[4] M. G. Ilieva, Olga Ormandjieva, “Automatic Transition of Natural Language Software Requirements Specification into Formal Presentation”, 2005.

[5] “IEEE Recommended Practice for Software Requirements Specifications”, in IEEE Standard 830-1998.

[6] Muhammad Ilyas, Josef Küng, “A comparative analysis of Similarity Measurement Techniques through SimReq Framework”, 2009.

[7] Naoaki Okazaki, Yutaka Matsuo, Naohiro Matsumura, Mitsuru Ishizuka, “Sentence Extraction by Spreading Activation with Refined Similarity Measure”, in The Florida AI Research Society Conference - FLAIRS, pp. 407-411, 2003.

[8] Silvia Salini, Ron S. Kenett, “Bayesian networks of customer satisfaction survey data”, in Journal of Applied Statistics - J APPL STAT, vol. 36, no. 11, pp. 1177-1189, 2009.

[9] Ronald D. Anderson, Robert D. Mackoy, Vincent B. Thompson, Gilbert Harrell, “A Bayesian Network Estimation of the Service-Profit Chain for Transport Service Satisfaction”, in Decision Sciences - DECISION SCI , vol. 35, no. 4, pp. 665-689, 2004.

[10] Anas Mahmoud, Nan Niu, “An Experimental Investigation of Reusable Requirements Retrieval”, in Information Reuse and Integration - IRI, 2010.

[11] Tao Zhang, Byungjeong Lee, “Complementary Classification Techniques based Personalized Software Requirements Retrieval with Semantic Ontology and User Feedback”, in Conference on Computer and Information Technology - IEEECIT, pp. 1358-1363, 2010.