An Approach for Automate Requirements Elicitation Using Use Case Models

Wai Wai Han, Khine Khine Oo
University of Computer Studies, Yangon
waiwaiwave@gmail.com, k2khine@gmail.com

ABSTRACT

Requirements elicitation is vital importance in system development process. While doing software system development, the stakeholders can't states real and nearly complete requirements because they might not know what they need or they have incomplete knowledge about the functionality of the intended system. To get requirements precisely from users, the software engineer must ask selective question using domain knowledge and requirements elicitation technique. In this paper, we present an approach to automate requirements elicitation system, which accepts stakeholders' requirements as questionnaires and checklists form that guide the user what their real requirements correctively. Our system automatically generates software requirements specification (SRS) and use case models. Due to the SRS, system analyst get more clear requirements as well as use case models give clear understanding of the proposed system.

Keywords
Requirements Elicitation, Requirements Specification

1. Introduction

Software requirements engineering (RE) is one of the early processes of software development life cycle that involves the process of discovering the system purpose, by identifying stakeholders, and their needs, and documenting these in a form that is amenable to analysis, communication and subsequent implementation [12] . Among the processes of requirement engineering, Software Requirements Elicitation may be the most important area of requirement engineering and possibly of the entire software processes because requirements are the misunderstood part of the system development and indefinite or ambiguous system requirements can cause serious problem for system developers [4][8]. It is generally accepted that errors produced at the requirements stage, if undetected until a later stage of software development and the requirements refinement can be very costly [10].

During requirements elicitation, all stakeholders (users, customers, developers, etc) are likely to be unfamiliar or incomplete knowledge about the functionality of the intended application [5]. This creates a barrier to motivate customers to state precisely what they need. The reason behind this difficulty is that the software engineer must ask selective question, mostly gained through experience, to be better elicit the true needs of the business and objectives of the stakeholders or to overcome this barrier [4].

In this paper, we present an approach to automate requirements elicitation using use case models.The purpose is to put an automate requirements elicitation system is implemented with the elicitation techniques using questionnaries and checklists, to support system analyst to further elaborate the system requirements that do not depend on a stakeholder's presence. This elicitation technique is simple and provides an efficient way to collect information from multiple stakeholders quickly. Moreover, the stakeholders' can write their requirements description as an input and then produce the whole system final results as software requirements specification (SRS view) and use case model to clear understanding of the requirements. The concept of natural language understanding use to transform the requirements description as a use case model that extract actors, use case and relationships from the description. Therefore every stakeholder between the system development team can communicate easily.

The remainder of this paper is organized as follows. In Section 2, background theory of the system describes. Section 3 explains how the architecture of automate requirements elicitation system will be implemented. Section 4 expresses implementation of the system. Conclusion and future work explain in Section 5.

2. Background Theory

The elicitation activity consists of gathering information about user needs, primarily from the users themselves. It is a process of helping the users to understand and articulate their requirements so they can make it to be known to the developers [7].

2.1 Questionnaries Elicitation Techniques

Questionnaries are one of the methods of gathering requirements in less cost [13]. Questionnaries reach a large number of people, not only in less time but also in a lesser cost. But the results extracted from the questionnaries should be clearly analyzed. The result from the questionnaries mainly depends on the two factors, firstly effectiveness and the design of the questionnaire dishonesty of the respondent. Secondly, a well-designed and effective questionnaries can be used to decide the actual user requirements objectives and the constraints [2]. A good structured questionnaire influences people to answer honestly thus making it possible to gather reliable results forms a large group of people. The data collected through questionnaries can be used to analyze the obtained results, both systematically and quantitatively [13]. The designing of questionnaries is a multi stage process and should be viewed accordingly.

The steps involved in designing and administering a questionnaries are [2] [13],

- The purpose of the survey should be defined
- The sampling group (respondents to be survey) should be decided
- Preparing and developing the Questionnaries
- Conducting the Questionnaries process
- Gathering and analyzing the results

Steps in arranging a questionnaries [11]

- The questions should be arranged well, so that general questions are followed by particular questions.
- Arrange the questions such that, easy questions comes first.
- Arrange the questions in an order of known to known
- Try to use closed format questions in the beginning
- The questions relevant to the main subject should be given high priority and stated at the start of the questionnaries.
- Avoid personal and intimate questions at the beginning

2.2 Use Case Diagram

Use case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of systems. Use case diagrams are used to visualize, specify, construct, and document the (intended) behavior of the system, during requirements capture and analysis [1]. Use Case views or diagrams describe the system from the User's point of view, and the various functionalities expected from the system to be developed and how user is going to interact with those features. Provide a way for developers, domain experts and end-users to communicate [6]. Serve as basis for testing. Use case diagrams mainly contain use cases, actors (the user), and their relationships (see Figure 1).

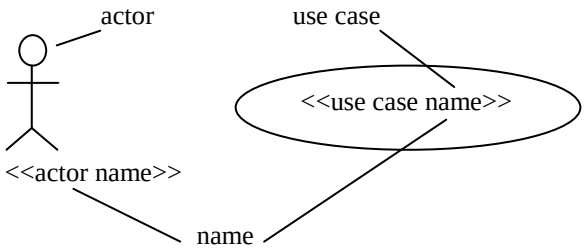

Figure 1. Actor and Use Case

2.3 Natural Language Processing

Natural Language Processing (NLP) is the engineering of systems that process or analyze written or spoken natural language. It is a subfield of artificial intelligence and computational linguistics. It studies the problems of automated generation and understanding of natural human language [3] [9]. There are different levels of knowledge for Natural Language Understanding. Some of knowledge is as follows:

- Phonetic and Phonological knowledge
- Morphological knowledge
- Syntactic knowledge
- Semantic knowledge
- Pragmatic knowledge
- Discourse knowledge
- World knowledge

2.3.1 Morphology

In morpheme-based morphology, a morpheme is the smallest linguistic unit that has semantic meaning. In spoken language, morphemes are composed of phonemes (the smallest linguistically distinctive units of sound), and in written language morphemes are composed of graphemes (the smallest unit of written language).

2.3.1.1 Type of Morpheme

The definition of a morpheme is "a minimal unit of meaning of grammatical function". There are two types or morpheme:

Free morpheme – can stand by themselves as single words. E.g., open and tour.

Bound morpheme – cannot normally stand alone, but which are typically attached to another form. E.g.,

re, -ish, -ed, -s. All prefix and suffix in English are bound morpheme.

3. System Design

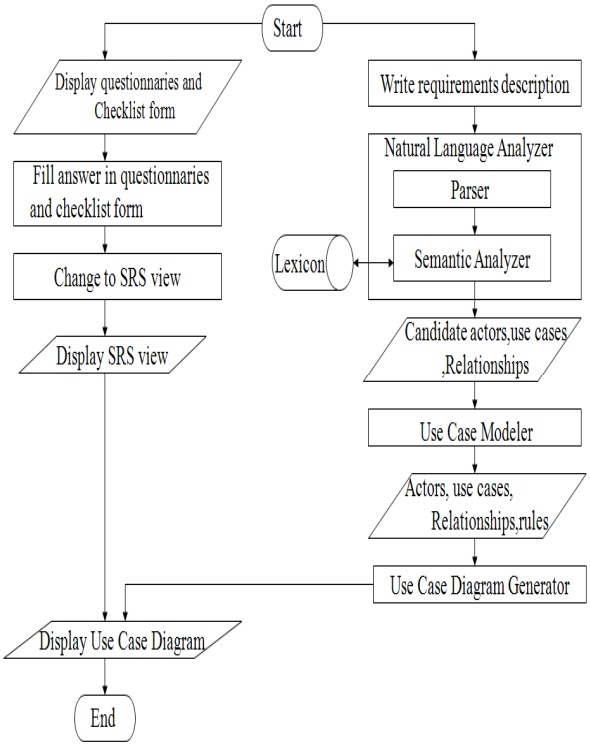

Figure 2. Overview of the System Design

The design of the automate requirements elicitation system is illustrated in Figure 2. In this system, there are two main parts. The first part is accepted stakeholder's requirements as questionnaries and checklist form. That is predefined question and answer by system analyst who uses questionnaries elicitation techniques. The stakeholders can fill their requirements in the form. Then the system transforms their input requirements into Software requirements specification (SRS) and use case diagram by using the rules of SRS and use case modeling to be clear understanding of their requirements.

At second part, the system will be accepted stakeholder's requirements or problem description as input sentences to produce use case model. Natural language analyzer performs tokenizing and these tokenize words are match with predefined data in lexicon to define candidate actor, use case and relationship. And then Use case modeler creates actors, use cases, relationships and rules as input of the use case diagram generator. Finally, the use case diagram will be generated by Use Case Diagram Generator. To elicit the requirements from stakeholders, they can not only fill their requirements in questionnaries and checklist form but also write requirements or problem description in this system.

3.1 Questionnaries and Checklist Form

An ideal questionnaries and checklists form includes the relevant information of the specific domain, for the stakeholder's who do not know what are their needs or what they want to be providing knowledge to suit the user's need. In this system, we will construct questions and answer table to elicit the requirements from the users automatically as questionnaries and checklists form that includes predefined questions and answer which is produced by system analyst using elicitation techniques. The data in the table can be changed upon the domain specific. In this system, we will present a template form to input the questions and answers of any domain easily.

3.1.1 Change to Software Requirements Specification (SRS) View

In this stage, the system will change the input requirements checklist data to Software Requirements Specification (SRS) view which is produced bases on their input checklist requirements and the rules of SRS.

3.1.2 Change to Use Case Diagram

At this stage, the system must be trained to extract actor and use case from stakeholder's requirements to draw use case diagram using the rules of use case modeling.

3.2 Capturing Requirements Description from User

Another way of getting requirements of our system is users can input their requirements description or problem description as a text document of the stake holder's requirements.

3.3 Natural Language Analyzer

The Natural Language Analyzer has a parser, which will be tokenizing the sentence; these tokens will then be analyzed by using semantic analysis with the aid of a business lexicon for identifying business specific terms.

The resulting information gathered from this phase will be candidate actors, candidate use cases and candidate relationships respectively and these are to be used by the next component called the Use Case Modeler.

3.4 Use Case Modeler

In this phase of the system, the use case rules and relationships will be created. These rules will then be used by the Use Case Diagram Generator as its input. The Natural Language Analyzer will populate the candidate actors, candidate use cases and the candidate relationships. The Use Case Modeler will then map these candidates in their correct and proper placing as well as creating the rules that will be applied.

The Use Case Modeler is the one responsible for the consistency of the use case diagram since the rules are created by this component. The Use Case Modeler will then store the correct and processed candidate actor, candidate use cases and candidate relationships as well as the rules generated into the corresponding fields.

3.5 Use Case Diagram Generator

In this phase of the system, the use case diagram will be created based on the rules, actors, use cases and relationships that were processed by the Use Case Modeler in the previous phase.

The Use Case Diagram Generator, with the use of the rules that were previously generated, will then be able to map out the correct placing and relationships of the use cases and their corresponding actors.

4. System Implementation

This system is implemented by using C# programming language and Microsoft SQL server 2005. In the system, software requirements specification (SRS) can be easily generated from the questionnaries and checklist form and produced use case models which are based on the stakeholder's input requirements. The output results of questionnaries and checklist as shown in Figure 3 and 4.

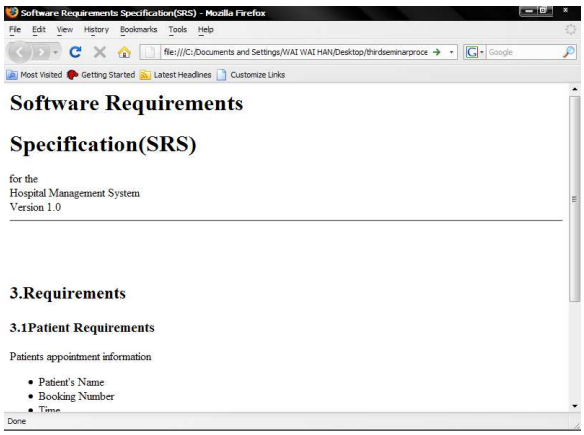

Figure 3. Software Requirements Specification

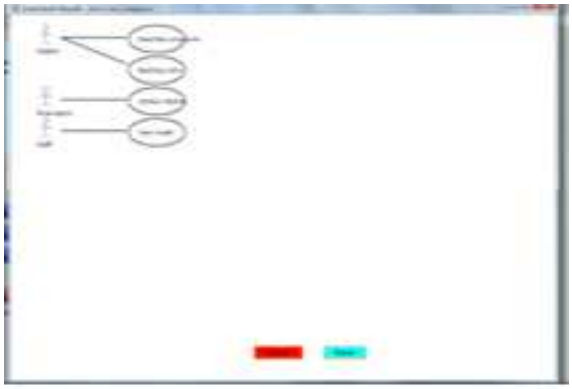

Figure 4. Use Case Diagram

Moreover, our system also produces use case models which is based on users input requirements description or problem description.

5. Conclusion

In this paper, we have presented an approach for automate requirements elicitation using use case models, which is based on the questionnaries and checklists of requirements elicitation techniques and concept of Natural Language Understanding.

The main idea of this approach is to accept users' requirements as questionnaries and checklists form or users' requirements description. Then produce final results as software requirements specification (SRS) and use case models. The use case diagram is important because it shows the requirements or functionalities of the system for clear understanding of their needs. From these views, other views like class diagram and sequence diagrams can be extended.

This system only emphasizes on the area of requirements elicitation and analysis phase and the system can be defined as only association, include and extend relationship between use cases if the user completely enters the requirements information. The generalization relationships between actors and use case can be extended in this system.

6. References

[1] Grady Booch, James Rumbaugh and Ivar Jacobson: The Unified Modeling Language User Guide

[2] J. Michael Moore, Frank M. Shipman: "A general introduction to the design of questionnaries for survey research", University of Leeds, online document. http://www.leeds.ac.uk/iss/documentation/top/top2.pdf

[3] Natural Language Processing

http://en.wikipedia.org/wiki/Natural_language_processing

[4] Neil W. Kassel and Brian A. Malloy, An Approach to Automate Requirements Elicitation and Specification, *Proceedings of the 7 th IASTED International Conference* SOFTWARE ENGINEERING AND APPLICATIONS November 3-5,2003, Marina Del Rey, CA, USA.

[5] B. Nuseibeh & S. Easterbrook, Requirements Engineering: A Roadmap, The future of software engineering, ACM Press, 2000, 37-46.

[6] Oberg, R. (2003). Applying Requirements Management with Use Cases, [online]. Available: http:// www3. software.ibm.com/ ibmdl/pub/software/relational/web/whitepapers/ 2003/apprmuc.pdf(March 26, 2004).

[7] Raghavan, Zelesnik, & Ford, "Lecture Notes of Requirements Elicitation, "1994: www.sei.cmu.edu/publications/documents/ems/ 94.em.010.html

[8] H. Reubenstein & R, Waters, The Requirements Apprentice: Automated Assistance for Requirements Acquisition, IEEE Transactions on Software Engineering, 17(3), March 1991, 226-240.

[9] E. Turban
"Expert System and applied artificial"
International Edition
ISBN-0-20-946565-6

[10] J. Van Buren & D. Cook, Experiences in the Adoption of Requirements Engineering Technologies, *CROSSTALK, The Jorunal of Defense Software Engineering*, 11 (12), December 1988, 3-10.

[11] Wai- Ching Leund:"how to design a questionnaries". University of East Anglia
http://homepages.inf.ed.ac.uk/mfelici/doc/questionnaries.pdf

[12] Zave, P., 1997, Classification of research efforts in requirements engineering: ACM Computer. Surv, v. 29, p. 315-321

[13] http://www.cc.gatech.edu/classes/cs6751_97_ winter/Topics/quest-design:"Questionnaries
Design ". Online document.