Research Journal of Applied Sciences, Engineering and Technology 6(23): 4393-4398, 2013
DOI:10.19026/rjaset.6.3442
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 05, 2013 Accepted: March 14, 2013 Published: December 15, 2013

Research Article

Structured Language Requirement Elicitation Using Case Base Reasoning

Marryam Murtaza, Jamal Hussain Shah, Aisha Azeem, Wasif Nisar and Maria Masood
COMSATS Institute of Information Technology, Wah Cantt, Pakistan

Abstract: Requirement elicitation is very difficult process in highly challenging and business based software as well as in real time software. Common problems associated with these types of software are rapidly changing the requirements and understanding the language of the layman person. In this study, a framework for requirement elicitation by using knowledge based system is proposed, which is very helpful for knowledge documentation, intelligent decision support, self-learning and more specifically it is very helpful for case based reasoning and explanation. Basically in this method requirements are gathered from Artificial Intelligence (AI) expert system from various sources e.g., via interviews, scenarios or use cases. Then, these are converted into structured natural language using ontology and this new problem/case is put forward to Case Based Reasoning (CBR). CBR based on its previous information having similar requirements combines with new case and suggests a proposed solution. Based on this solution a prototype is developed and delivered to customer. The use of case-based reasoning in requirements elicitation process has greatly reduced the burden and saved time of requirement analyst and results in an effective solution for handling complex or vague requirements during the elicitation process.

Keywords: Case Base Reasoning (CBR), expert systems, ontology, requirements elicitation, requirement analyst

INTRODUCTION

Requirements elicitation is one of the most crucial phases in the software development life cycle. Although, it is the most important phase but it is often misled due to lack of understanding or cultural differences among the developers and stakeholders. This becomes the main reason for project risks or in other words failure to deliver the software in accordance with user needs. Obviously, if the software being developed does not satisfy the customer it will not only be havoc on customer organization but it will also have bad influence on the reputation of the organization developing the project.

These problems arise due to incomplete or ambiguous requirements, which results in great problems for requirement analyst to inter-relate the information obtained via various elicitations techniques. Various studies report that improper requirement elicitation may result in enormous failures of projects (Davis *et al*., 2006; Breward *et al*., 2009). Poor elicitation of requirements not only cause project failures but it's also a major reason for cost overrun, when requirements change and hence handling these new requirements increase cost of overall project (Urquhart, 1999). Many techniques for requirements elicitation have been proposed that try to bridge the gap of understanding between stakeholders and requirement analyst and each has its own pros and cons.

Another way to solve the problems of requirements elicitation is to add machine-learning capabilities during the software development process. Adding this facility reduces the workload for the organization developing the software. Different machine learning techniques can be induced during the software development phases like Neural Networks (NN) (Haykin and Network, 2004) can be useful for classification, Case Based Reasoning (CBR) is also handy for finding solutions based on previously solved problems, each such problem is referred to as a case (Watson and Marir, 1994). A review of such techniques is presented in Singh *et al*. (2007).

Many techniques are employed for requirements elicitation, which are broadly classified into four categories. First, is the conversational method (Zhang, 2007), which includes techniques like brain storming, focus groups and via interviews (Goguen and Linde, 1993). Such methods are easy to conduct and involve direct involvement with stakeholder. During this the requirement analyst has to be very focused and attentive to note all information provided by the customer.

Second category belongs to observational approaches, which require the understanding of customers working environment as in ethnography (Wongthongtham *et al*., 2009). This is useful because in this way the analyst can check for himself, how the people work and their regular activities in that application domain for which the software is being developed. Studies related to ethnography are stated in Nuseibeh and Easterbrook (2000).

Corresponding Author: Jamal Hussain Shah, COMSATS Institute of Information Technology, WahCantt, Pakistan

Third category lists the analytical approaches for elicitation of requirements. In these techniques instead of gathering information from user or their work place, requirements are obtained by similar problems already solved, as described in Cybulski and Reed (2000). Such methods are helpful to get a clear understanding of the application domain.

Finally the last category deals with syntactic methods for elicitation, in which the customer/stakeholder together with the requirement engineer communicate in an informal way by relating the requirements to everyday scenarios. Therefore, they agree on the commonalities of requirements extracted from them. Scenario based approaches are described in Leffingwell and Widrig (2000) and Sharp *et al*. (1999).

Liao (2010), conceptual model based on VCA (Value Chain Analysis), was investigated to overcome the cognitive limitations that cause analysts to gather inadequate and inaccurate requirements. On the other hand in Jiang-ping *et al*. (2009) used the knowledge conversion to provide the solution knowledge dissymmetry between clients and developers. He implemented the SECI Spiral Model to software requirement elicitation. While in the context of Ramsin and Paige (2010) used an iterative method for the specification of requirements of an SDM (Software Development Methodology). He performed the method as the analysis phase which ultimately led him to the foundation of target methodology.

Further in Liu and Lin (2008) developed an automated elicitation system which is based on the knowledge-base of user requirements. In this methodology, by gradually improving the original requirements description, the quality and efficiency of requirements elicitation process can be improved as well. This methodology assumed that a complete and rigor knowledge base of a specific domain is in place, inquiry cycles are formed into the problem of decision-making.

Later in Runde *et al*. (2009) applied Semantic Web Technologies for the purpose of optimizing the engineering process in building automation.

While in Mishra *et al*. (2008) implemented Supply Chain Management for the real life project of a small scale software development organization where the requirements were volatile. It overcomes the problem if the requirements are not consistent.

According to Wei *et al*. (2007), HSI (Heavyweight Semantic Inducement) for the analysis of gathered requirements was used. By semantic match rules, this HSI automatically analyze the requirements whether to delete or to keep a specific requirement for developing consistency in requirements elicitation and analysis.

While in Tom and Sitte (2009) used REFUSS (Requirement Elicitation of Future Users by Systems Scenarios) that is used to collect the requirements of future user by relating process knowledge of current systems with future scenarios. It uses needs base vision and hence it overcomes the failures generated by technology based vision.

Beside the above-mentioned methods for requirement collection, there are many other techniques described by various authors. These techniques are not solutions for a particular problem but it depends on the application for which the software is to be developed.

But due to hindrances during the requirements elicitation phase requirements are not properly extracted and the analyst has to go through difficult time in-order to structure the requirements in proper format. To solve this difficulty and to save time for the analyst this study presents a machine learning approach for solving problems that arise during requirement elicitation phase. Detailed description of the proposed method is described in later section.

PROPOSED TECHNIQUE

The proposed technique is used to solve the problem that arises during the requirement elicitation is based on subfield of artificial intelligence i.e., Natural Language (NL) understanding and Case Based Reasoning (CBR). The problems that are faced by the requirement analyst may arise due to lack of understanding, scope or changing requirements. To overcome the problem of natural language and to reduce the burden and save time of requirement analyst a technique is proposed in this study. This technique is comprised of two parts. In the first part, expert systems gather requirements using various techniques like scenarios, interviews, use cases, open and close ended questionnaire and viewpoints. After collecting requirements in the form of natural language ontology is used that converts into structural language. In the second part, case based reasoning is used that follow knowledge based system that containing inference rules and used for future work out on similar type of system.

Structured language requirement elicitation using AI techniques: Natural Language understanding is a very critical issue that is faced by various requirement analyst as well as different stakeholders. These problems may arise because different stakeholders specify the same requirements in different ways, which leads to misunderstanding and is very ambiguous for requirement analyst to interpret.

In this study a technique is proposed to identify the requirement using various AI techniques. These AI techniques are very helpful and overcome the ambiguity for system analyst. The AI techniques, which are used to gather requirements, are Knowledge based and Expert systems. Knowledge base system is very cooperating and is used as future reference during knowledge documentation, intelligent decision support, self-learning and is more specifically for reasoning and explanation. The types of Knowledge based systems

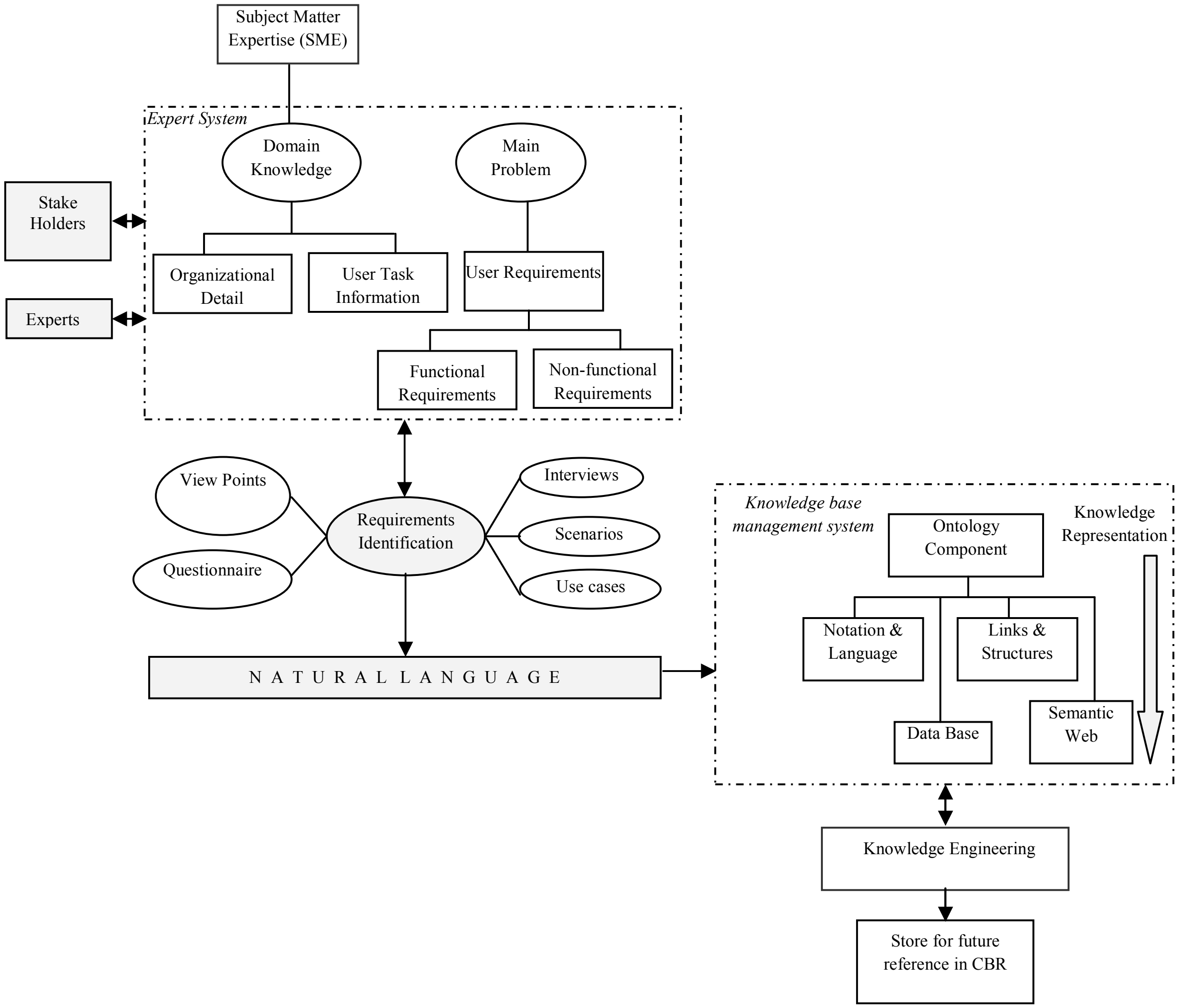

Fig. 1: Structured language requirement elicitation using AI techniques

such as expert systems, case based reasoning and neural networks but we prefer expert system for requirements identification and case based reasoning as reference for similar type of system.

Although there are many ways to gather and extract requirements such as interviews, scenarios, or use cases, surveys etc. but they require a lot of time to manipulate in a proper format. But if the requirements are gathered using expert systems and knowledge base management system then it is appropriate for system analyst and their lot of the time is saved during requirements elicitation as shown in Fig. 1. A detailed description of converting a set of natural language into detailed structure language is as follows:

- First of all experts gather information from external environment and different stakeholders. Experts collect information about particular application domain and current problem from different entities
- Subject Matter Experts (SME) are responsible for gathering domain knowledge. SME are the one who have domain expertise, so they get the organizational requirements as well as user task information (that contain current problems associated with area knowledge) using their particular domain knowledge
- Experts assemble the domain knowledge and current problem simultaneously. Current problems are collected from user requirements, which contains functional as well as non-functional requirements
- These experts identify above requirements using different techniques like view points, open or close ended questionnaire, scenarios, interviews, use cases etc. But again the problem arises due to natural language, which may create ambiguity or misunderstanding for system analyst
- To handle the problem of natural language and convert it into structured form another AI technique is used i.e., Ontology. Ontology converts the English sentences into formal structured form using its own components. These components are

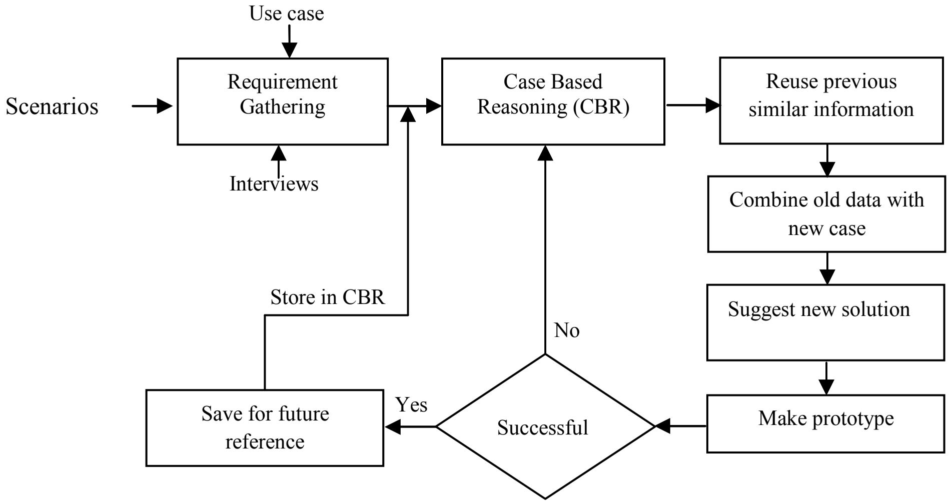
Use case
Scenarios
Requirement Gathering
Interviews
Case Based Reasoning (CBR)
Reuse previous similar information
Combine old data with new case
Suggest new solution
Make prototype
Successful
No
Yes
Save for future reference
Store in CBR

Fig. 2: Flow chart of the proposed system

classes, attributes and set of rules, relations, events and restrictions. Now these components are converted into structured form by using knowledge representation i.e. notation and languages, ontology languages, links and structures, databases, semantic webs and storage and manipulation etc. Now, this can be called as knowledge based system, which contains detailed knowledge about the system

- At this point, the system analysts have detailed knowledge about system requirements. When these requirements are fed into the computer system, a discipline is developed that involves the integration of knowledge into computer by adopting the process of knowledge engineering. It refers to the building, preserving and development of knowledge based system. Finally it can be stored into case based reasoning for future reference

So, all these problems put great responsibility on the requirement analyst in a sense that he has to maintain a record of all the information retrieved from the customer. Whether, that information is a goal/objective/functional requirement or a system requirement. After analyzing the obtained information, analyst has to decide which information is relevant and which to ignore. But again this would only be possible if the interviews were conducted in a proper format, with right kind of question set in front of the customer. Even if a structured approach is followed it requires a lot of time to sort the requirements and make appropriate decisions.

To avoid this entire hassle, case base reasoning is introduced during the elicitation process. Flow chart of the proposed solution is described in Fig. 2.

The detail description of the process presented in Fig. 2 is given as under:

- 1. Firstly, the phase of requirement collection through any source (interviews, scenarios etc.) is conducted for the stakeholders.
- The obtained information is passed to CBR for classification i.e. new problem/case is fed to CBR block where the new case is combined on the basis of similarity with previously solved case
- Based on this reuse, a solution is proposed i.e., the actual methodology is defined which is to be used for further in design phase
- After obtaining the solution, a prototype is made and delivered to the customer. If satisfactory response is obtained from the customer, further development is continued
- Also the new solutions based on the requirements collection are stored for future references, i.e., if similar case occurs in future this can be reused to solve a particular problem
- Alternatively, if proposed solution does not satisfy the customer, refinement is made in the solution
- Ourmajorobjective of introducing the case base reasoning during the requirement elicitation process is that, it has reduced a great burden of the analyst and hence saved a lot of time for which requirement analyst had to consume while trying to understand and relate the given requirements from the customer

Case study-Student Information System (SIS): an ERP system for universities: For this elicitation model, we developed a Student Information System (SIS) and deployed it at an XYZ Education organization. This system was expected to work in a right sense for about 4 years. The initial system was developed by using Waterfall Process Model as it was an in-house development and all the experienced and concerned people were in the software development team. The system was tested by the team-members and found that it had been giving up to 68% correct results (over all completion).

But after the complete deployment of the Student's Information System at the education institute, the users of the system i.e., faculty members and students started to demand for an enhanced version of Graphical User Interface. In the beginning, the GUI was kept simple just to know the response as if people would like to use online system for their course registration; attendance details, marks details, fee payment and many other features were also included. So, a deep concentration was given to the development of functionality and structure of the system rather graphical user interface. But after the users of the system asked for a good interface, our team started to work on it. For this purpose, we decided to choose Prototyping because it was an interactive system. With the successful completion of interface design phase, it was again made to use by the users i.e., students and faculty of the university.

Later with further usage of the system, the user's complaint that there are errors in the system for some cases like sometimes it shows wrong courses to be registered in a semester or sometimes the system registers the same course of the student repeatedly. Sometimes, it happened that students submitted an online fee, but it was not notified to the accounts office. These issues were causing troubles for the students and they had to visit the IT solutions department frequently. To fix these bugs, our team started making amendments in the Student's Information System. This time our development team selected the exploratory development procedure. Therefore, a team of faculty and a team of students were involved in the development process and the system was continuously tested by the testing team. Each and every bug identified by the user in the previously deployed system was fixed by the development team. In this case, the system was tested by module testing followed by an integration testing. It showed 72% correct results.

In the end, further improvements were performed by the development team. It was researched by the Research and Development (R&D) Department to add some other modules to the Student's Information System so that the teachers and students can be provided with a more interactive platform. This was done by the above proposed requirement elicitation technique and the results were obtained after carrying out the modular testing and an integration testing. These results were far better than the results which were obtained from previous systems. It gave 90% correct results.

CONCLUSION

Different AI techniques such as knowledge based system; ontology and case based reasoning are used during requirements elicitation which can minimize the problem of natural language understanding as well as greatly reduce the burden and save the time of requirement analyst. Natural Language is the most critical issue that can be reduced by using ontology, which converts the natural language into structured language. On the other hand Case Based Reasoning approach is used during requirement elicitation process. The use of CBR during the requirement phase is a vital step to handle complex and/or vague requirements that are often given by the stakeholders. As a result, requirement analyst has to go through a great hassle in order to understand what exactly the customer is requiring. So, the proposed solution can greatly minimize the overhead by facilitating the requirement analyst.

REFERENCES

Breward, M., M. Head and K. Hassanein, 2009. Consultants experience of requirements elicitation conversations: An empirical model. Proceeding of the ECIS. Paper 33.

Cybulski, J. and K. Reed, 2000. Requirements classification and reuse: crossing domain boundaries. Proceeding of the International Conerence on Software Reuse: Advances in Software Reusability, Springer-Verlag London, pp: 1-8.

Davis, C.J., R.M. Fuller, M.C. Tremblay and D.J. Berndt, 2006. Communication challenges in requirements elicitation and the use of the repertory grid technique. J. Comp. Inform. Syst., 46(5): 78.

Goguen, J.A. and C. Linde, 1993. Techniques for requirements elicitation. Proceeding of IEEE International Symposium on Requirements Engineering, 1993, Comput. Lab., Oxford Univ., UK, pp: 152-164.

Haykin, S. and N. Network, 2004. A comprehensive foundation. Neural Networks, Vol. 2.

Jiang-Ping, W., H. De-Yi and D. Wan, 2009. Knowledge conversion in software requirement elicitation. Proceeding of the 1st International Conference on Information Science and Engineering (ICISE), Sch. of Bus. Adm., SCUT, Guangzhou, China, pp: 2328-2331.

Leffingwell, D. and D. Widrig, 2000. Managing Software Requirements: A Unified Approach. Addison-Wesley, Boston.

Liao, H., 2010. Requirement elicitation of enterprise informationization from view of VCA. Proceeding of the 6th IEEE International Conference on Networked Computing (INC), Sch. of Software, JiangXi Univ. of Finance&Econ., Nanchang, China, pp: 1-6.

Liu, L. and L. Lin, 2008. ARED-CK: An automated requirements elicitation approach based on decision-making with complete knowledge. Proceeding of the 1st IEEE International Workshop on Managing Requirements Knowledge, Sch. of Software, Tsinghua Univ., Beijing, pp: 47-52.

Mishra, D., A. Mishra and A. Yazici, 2008. Successful requirement elicitation by combining requirement engineering techniques. Proceeding of the 1st IEEE International Conference on the Applications of Digital Information and Web Technologies, Dept. of Comput. Eng., Atilim Uni., Ankara, pp: 258-263.

Nuseibeh, B. and S. Easterbrook, 2000. Requirements engineering: A roadmap. Proceeding of the Conference on the Future of Software Engineering, ACMNew York, pp: 35-46.

Ramsin, R. and R.F. Paige, 2010. Iterative criteria-based approach to engineering the requirements of software development methodologies. IET Software, 4(2): 91-104.

Runde, S., A. Fay and W. Wagner-Otto, 2009. Knowledge-based requirement-engineering of building automation systems by means of semantic web technologies. Proceeding of the 7th IEEE International Conference on Industrial Informatics, Inst. of Autom. Technol., Helmut-Schmidt-Univ. Hamburg, Hamburg, Germany, pp: 267-272.

Sharp, H., C.W.A. Finkelstein and H. Galal, 1999. Stakeholder identification in the requirements engineering process. Proceeding of the 10th IEEE International Workshop on Database and Expert Systems Applications, Centre for HCI Design, City Univ., London, UK, pp: 387-391.

Singh, Y., P.K. Bhatia and O. Sangwan, 2007. A review of studies on machine learning Techniques. Int. J. Comp. Sci. Secur., 1(1): 70-84.

Tom, M. and J. Sitte, 2009. Future user requirement elicitation for technology investment: a formal approach. Proceeding of the IEEE IEEE International Conference on Systems, Man and Cybernetics, Sch. of Inf. Technol., CQUniversity, Brisbane, QLD, Australia, pp: 2116-2121.

Urquhart, C., 1999. Themes in early requirements gathering: The case of the analyst, the client and the student assistance scheme. Inform. Technol. People, 12(1): 44-70.

Watson, I. and F. Marir, 1994. Case-based reasoning: A review. Knowl. Eng. Rev., 9(4): 327-354.

Wei, L., H. Ke-Qing, W. Jiang and P. Rong, 2007. Heavyweight semantic inducement for requirement elicitation and analysis. Proceeding of the 3rd IEEE International Conference on Semantics, Knowledge and Grid, Wuhan Univ., Wuhan, pp: 206-211.

Wongthongtham, P., E. Chang, T.S. Dillon and I. Sommerville, 2009. Development of a software engineering ontology for multisite software development. IEEE Trans. Knowl. Data Eng., 21(8): 1205-1217.

Zhang, Z., 2007. Effective Requirements Development-A Comparison of Requirements Elicitation Techniques. University of Tampere, Finland.