Requirements Elicitation Techniques Selection Based on Taxonomy of Project Type

Getu Beshah
getubeshah@gmail.com

Mesfin Kifle
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

The success of a system development is greatly dependent on the quality of the requirements. The quality of the requirements, in turn, is highly affected by the type of elicitation techniques that are employed during the requirements elicitation process. Requirements elicitation is a complex process involving many activities and it is subject to a large degree of error, influenced by key factors ingrained in selection inappropriate elicitation techniques.

To alleviate this problem, nowadays knowledge based selections of software requirements elicitation techniques approaches come in to existence, and are tailored to use many elicitation techniques considering the techniques suitability. The study is founded on the previous works in knowledge based systems, evaluating and building upon their efforts and extended the knowledge based approach by considering taxonomy of project characteristics. Its aim is to fill this gap which provides necessary support for the analyst by taxonomy of the software project types.

The proposed method for requirements elicitation techniques selection provides a means to select suitable elicitation techniques based on the project type and related knowledge base. The method is evaluated using a prototype and a case study to demonstrate its feasibility and effectiveness.

Keywords: Taxonomy; Project Type; Project Domain; Project Attributes; Elicitation Technique

1. Introduction

The success rates of software projects are remaining too minimal over the years due to various technical and managerial issues [1]. Requirement Engineering is a key issue for the development of software systems with the responsibility for maintaining the requirements of a system. Correct elicitation, specification, and validation of user needs are becoming more and more crucial as the ultimate measurement for systems quality is the degree of user satisfaction.

Even if a lot of methods exist to perform the requirement gathering process, they are still facing problems in gathering the requirements due to lack of knowledge on how to select appropriate techniques. According to [2], lack of support for flexible, reliable, and adaptable requirement engineering processes is a major issue that prevents organizations from improving their requirement engineering practices. Therefore, improving this process will considerably increase developers' productivity and quality of the software products [3].

2. Background

Requirements elicitation, which is one of the requirement engineering activities, is a complex process involving many activities with a variety of available techniques, approaches, and tools for performing them. The multidisciplinary nature of requirements elicitation only adds to this complexity. It is subject to a large degree of error, influenced by selection of inappropriate elicitation techniques [4]. Poor execution of elicitation techniques mostly contributes to project failure. Hence improving the industry performs elicitation could have a dramatic effect on the success record of the industry [5].

Most models of requirements elicitation focus on specific methodologies or techniques and specific process models [5]. As indicated in [6], some analysts think that, one technique is applicable to all

situations, but one technique cannot possibly be sufficient for all conditions.

Nowadays using the specific methodology or technique is tailored to use many elicitation techniques considering the techniques suitability. Hence, knowledge based selections of software requirements elicitation techniques approaches come in to existence. The author in [3] developed a framework for selecting appropriate elicitation techniques that consider selection factors that are derived from the requirement type, project environment, and the problem domain constraints. Many papers have been written on knowledge based requirements elicitation techniques selection and this study is founded upon related literature, evaluating and building their efforts; and extending the knowledge based approach by considering taxonomy of project types. Hence, the main contribution of this paper is to fill the gap of Requirement Elicitation Techniques Selection process based on taxonomy of project type, which aims to provide necessary support for the analyst. The proposed method provides a means to select suitable elicitation techniques.

3. Related Work

The authors in [5] proposed a mathematical model of the requirements elicitation process that shows the role of knowledge in its performance. They did a model for two Knowledge-Intensive software development processes. These are process of requirements elicitation, and selection of an appropriate elicitation technique.

In their work, they stated the role of knowledge in performing selection. The knowledge base includes (a) the current problem, solution, and project characteristics, (b) the awareness of which requirements are known and which are still to be determined, and (c) relationship of the current problem, solution, and project characteristics. Furthermore, they stated the requirements to the selection of an elicitation technique.

For our study, we focused on one of the two knowledge intensive software development processes model, which is elicitation techniques selection as shown in Figure 1 [5].

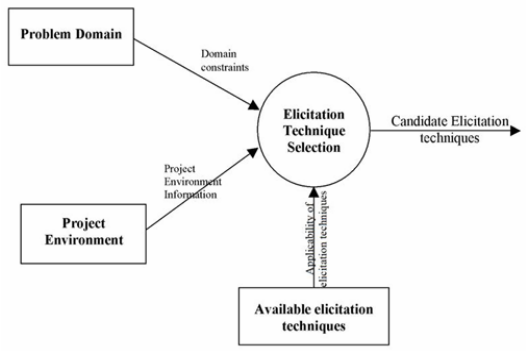

Figure 1: Elicitation Technique Selection

As shown in Figure 2, the approach doesn't deal with taxonomy of problem, solution, and project characteristics, but the authors noticed it as future work.

The author in [3] proposed a framework which assists analysts in the selection process of elicitation techniques that fit the goal of elicitation session, the project environment and the problem domain. The proposed framework has three steps. It takes the goal of an elicitation session, information about the project environment and the problem domain and generates list of suitable elicitation techniques in their priority of applicability to a given condition. But the research did not consider taxonomy of the project characteristics for easy support for the analyst.

The author in [1] proposed a knowledge based software management model selection approach for software and project management and development. The aim of this work is to help practitioners in making better selection decision of appropriate software development and management models. It is focused on knowledge based system on the area of project management models selection.

The research in [7] focused on requirements elicitation techniques selection. According to this research, requirement analysts who have extensive experience seem to have the ability to select appropriate elicitation techniques on a regular basis. But most practicing analysts have less experience and are more journey man than expert analysts. Thus, in their paper, they suggested that if the average analyst's ability to select elicitation techniques were improved, it would most likely be possible to improve their record of successful software products.

Finally, they suggested that mechanisms should be created to formulate the experts experiences so that less experienced analysts could use it easily because less experienced analysts do not know how and when to apply elicitation techniques. The authors discussed the results of in-depth interviews with the world's most experienced analysts. However, they had interviewed only a small sample of experts.

characteristics on the former studied knowledge based system. It is used for the analyst to select suitable requirements elicitation techniques for a given software project, that is better for complex system and increases efficiency. The model listed out project types and then project domains and the project attributes and finally provides different suitable elicitation techniques.

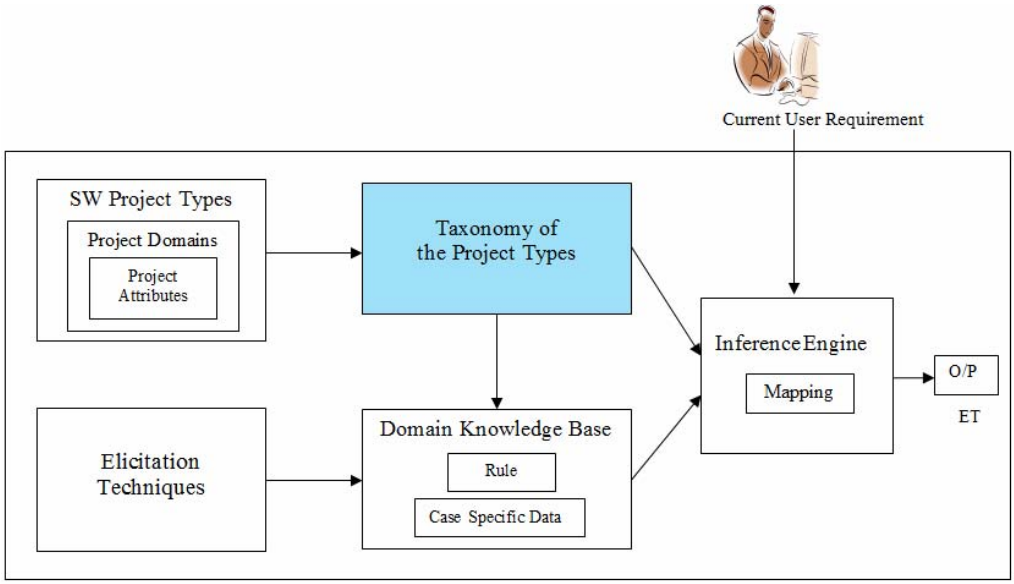

Figure 2: The structure of Knowledge based Requirement Elicitation Techniques Selection Model

4. The Proposed Solution

The proposed model is shown in Figure 2, by including the component of taxonomy of project

4.1 Software Project Problem Space

According to [6], a software project environment is partitioned in to two domain spaces, which are problem space and solution space. A project characteristic, like project type, project domain and project attributes are considered as problem space and requirement engineering techniques are included in solution space. We used this concept for the subsections of software project type, project domain, and project attributes. These software project characteristics are applied to group as project type, project domain, and project attributes for our research work of taxonomy based requirements elicitation techniques selection.

The components of the model are software project problem space, elicitation techniques, taxonomy of the project type, domain knowledge base, and the components of inference engine.

4.1.1 Software Project Types

The classification of software maintenance types in [8, 9]] are the basis for classifying the software project for this research work.

The work in [9] classified the maintenance software into four parts such as Corrective, Adaptive, Perfective, and Preventive Maintenance. Taking this concept, we considered each maintenance part as software project type. Additional project type, original system, taken from [10] as defined as the project types, and was necessary for our research work due to relevant project type for our research design. Generally, five project types are defined in this research.

4.1.2 Software Project Domain

The project domain is referred to the knowledge about a specific field of interest of applications such

as health, education, entertainment, finance, etc. and each type of project domain is characterized by its project attributes. Most researches concentrated on generic solutions for software problems. However, when we come to specific domains, they have their own special nature. Hence, there is a need to take domain considerations in order to manage the complexity of the software requirement engineering process for selecting suitable techniques [3].

4.1.3 Software Project Attributes

When developing requirement engineering process model for a given software project the first step is to identify the attributes of the software project. For this research, 16 software project attributes are defined. Based on previous research that indicated in [6], a comprehensive set of project attributes need to be developed that is suitable for the vast majority of software projects and can be linked to the characteristics of requirement elicitation techniques. This link helps to select suitable elicitation techniques based on the attributes of the project.

4.2 Elicitation Techniques

In this work, 25 elicitation techniques have been identified by referring survey results and from literature [3, 4, 11, 12, 13]. The survey is conducted using a questionnaire and eight local private software development companies responded. The result shows the most common elicitation techniques used are Interview, Document Analysis, Observation, Prototype, Questionnaire, JAD, Requirement Reuse and Model-based. We considered all these techniques for this research.

4.3 Taxonomy of the Project Types

The software project type is designed as a tree structure method of taxonomy. We classified the project type considering taxonomy with the intent of easy to search the problem space. The software type, project domain, and software project attributes are grouped together into the taxonomy on the concept of a 3-level view to make it easy to get and make usable for the analyst.

Under each project type, there are different project domains. For this research, we considered health domain knowledge as it had been studied in [3], and has been used as input source for taxonomy of the model proposed.

4.4 Domain Knowledge Base

The domain knowledge is referred to the knowledge about a specific field of interest of an application such as health, education, finance, etc. and health domain is chosen as a sample representative. The reason for the selection of this domain is the availability of documents in the health domain area and the nature of the health domain and type of applications that are in use in the domain studied in [3].

4.5 Inference Engine

An inference engine is a program that conducts the reasoning process for solving problems based on the knowledge contained in the knowledgebase. It interprets the knowledge and derives a solution of the problem. It is the processing program in the knowledge based system that derives a conclusion from the cases and rules contained in the knowledgebase.

4.5.1 Assumption

The current requirements of a user or the current problem definitions have equivalent weights with the domain knowledge of past problem definition for selection of the suitable techniques process. The past domain knowledge solutions can work in new situations and they still help when new solutions are being developed.

4.5.2 Technique Suitability Assessment Matrix Model

When we select the attributes options value, the inference engine maps the attributes options with the techniques, and when we request the recommended techniques, it provides the list of elicitation techniques according to the Technique Suitability Assessment result that start from the top max value to min. When we set the number of elicitation techniques, we get the requested number of techniques, or else it will give us the default number of techniques, 5.

Table 1: Technique Suitability Assessment Matrix Model

PA	ET				
	Option	T_1	T_2	…	T_N
PA_1	O_1	C_{111}	C_{121}	…	C_{1N1}
	O_2	C_{112}	C_{122}	…	…
	…	…	…	…	…
	O_N	C_{11N}	…	…	C_{1NN}
PA_2	O_1	C_{211}	C_{221}	…	C_{2N1}
	O_2	C_{212}	C_{222}	…	…
	…	…	…	…	…
	O_N	C_{21N}	…	…	C_{2NN}
…	…	…	…	…	…
PA_N	O_1	C_{N11}	C_{N21}	…	C_{NN1}
	O_2	C_{N12}	C_{N22}	…	…
	…	…	…	…	…
	O_N	C_{N1N}	…	…	C_{NNN}

Table 1 indicates the mapping of the "project attributes option values" and the "elicitation techniques" to get the "suitability of the techniques".

The function used in the process is:

$F = PA \times T \times O \rightarrow C$

where F = Function

O = Option

T = Techniques

C = Cell

PA = Project Attributes

To show the application, the following function is taken arbitrarily. It is done for two columns of T1 and T2. In this case we calculated the cells column wise. When we calculate the cells in a column, we have to sum all cells in the corresponding project attributes and then we get the value of the elicitation techniques in a single column.

F = T1xPA1xO1+ T1xPA2xO1 +T1xPA3xO1 …T1xPANxO1

C111 + C121+ C131… C1N1 = Total C1 in column T1 (1)

F = T2xA1xO1 +T2xA2xO1 + T2xA3xO1 … T2xANxO

C211 + C221 + C231 … C2N1 = Total C2 in column T2 (2)

To show the application, compare the above functions (1) and (2) and we can see how selection of the best suitable elicitation techniques is provided by the engine. Now to get the best elicitation techniques, we need to compare the values C1 and C2 and we take the larger value as a more suitable elicitation technique than the lesser one.

4.5.3 Inference Engine Performance due to Taxonomy

Basically the Inference Engine performs two things due to the taxonomy of the project characteristics. The first and main thing is that it is used when the system is complex. That means, when the system is comprised of different domains, the Inference Engine recommends different related elicitation techniques according to the specified domains. For example, ethio telecom launched a new system in 2012, which is used for three domains of Human Resource, Finance, and Procurement. In the future, the company will add more other domains.

To develop this type of complex system, the taxonomy based model recommends different elicitation techniques for different domains of a system. For example, for Human Resource the method recommends related elicitation techniques, and for the rest two domains it recommends their other related/appropriate elicitation techniques.

The second importance of the work is in terms of performance. Efficiency of the Inference Engine is improved since the taxonomy of the project types minimizes the searching space. It is more compact, thus it requires less storage space. This is so because taxonomy does not contain terms for each compound concept. It is more scalable and can be maintained more easily. The addition, deletion, and updating of terms are easier and can be implemented more efficiently. It is better suited for the classification of strong foundation for creating new groupings of project characteristics.

The inference engine dynamically generates navigation trees that are suitable for browsing the project characteristics. Some search engines (e.g., Google) employ taxonomies in order to enable limiting the scope of searches. The search engine will compute the degree of relevance with respect to only of those pages that fall in the category. Clearly, this enhances the precision of the retrieval and reduces the computational cost [14, 15].

According to [16], taxonomy of the project types reduces duplication of items of similar type. Furthermore, it presents the cascaded inference algorithm for improving the efficiency of inference. According to [17], the ability of a cluster to

categorize by assigning items to groups provides more efficient or more effective retrieval. On the related concept in [18], for faster information retrieval and a better classification of knowledge, taxonomy is very much essential; it is used easy navigation and searching.

5. Prototype

The main objective of the prototype is to demonstrate the model, and also, it serves as a preliminary work for future development of the Elicitation Techniques Selection System tool, based on Taxonomy of Project Type. The Model displays list of software project characteristics, such as project types, project domains and list of the project attributes, list out each project attributes options value and finally lists out the recommended suitable elicitation techniques.

To set the default number of elicitation techniques, we used the average of the elicitation techniques survey data that is conducted in local private software development companies, so that the optimal default number of elicitation techniques is 5. But if we need more or less than the default one, the model can do it.

The prototype provides the following features:

- The type of interface of the model is window based.
- It provides list of software project characteristics, such as project types, project domains and list of the project attributes, lists out each project attribute options value and finally lists out the recommended suitable elicitation techniques.
- The user can see all information in one interface such as the selected project domain, project attributes and selected project attributes option value and the recommended elicitation techniques.
- It provides the interface to enter the user requirements of the project attributes with the corresponding option values. The Inference Engine then analyzes these with the knowledge and finally yields the recommended best suitable elicitation techniques.
- It provides interface for the addition, deletion, and modification of software project characteristics for elicitation techniques, project types, project domains, project attributes and project attributes option values and the Matrix.

The project type interface, Figure 3, provides the functionality of the project types. When a user selects a specific project type, it leads next to selected specific project type interface. Also it provides the access to add, delete, and modify when entering to maintenance tap and it provides termination access.

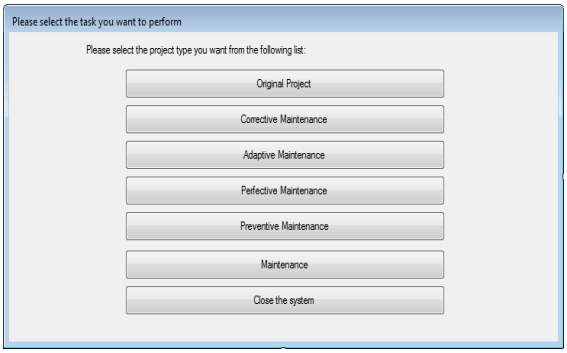

Figure 3: Project Type Interface Design

When a user selects one of the project types, the prototype provides the main multitask interface (see Figure 4). The main multitask interface provides list of project domain, list of project attributes, list of project attributes option values, and provides list of selected recommended suitable elicitation techniques. For the process of selection of suitable election techniques, it provides the functionality to accept the requirement information from the user and finally, after accepting the request, it lists out the suitable elicitation techniques. The prototype provides maintenance features too. When we select the maintenance tab, it will open a new interface window which has save, update, and delete tabs for maintenance purposes for all project types, project domains, project attributes, for matrix and for elicitation techniques.

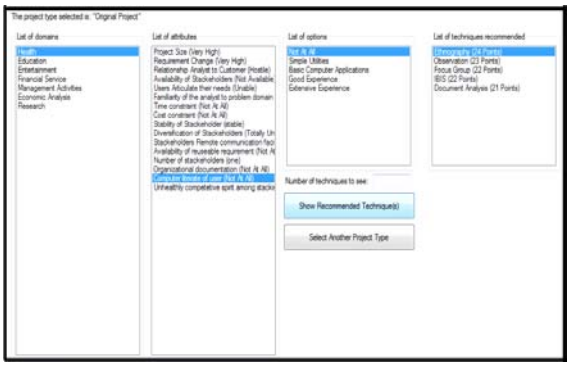

Figure 4: Main Multitask Interface Design

In order to show the workability of the prototype, clinical data management system, which is used as an example to illustrate the effectiveness of the system in [3], is selected.

Generally the result shows that, we can get suitable elicitation techniques using this knowledge based selection of appropriate software elicitation techniques based on taxonomy of project characteristics.

But the results obtained in this case study cannot necessarily be generalized and cannot guarantee that similar result would be achieved as in [3]. Because, in our case, the prototype model is able to recommend suitable elicitation techniques with an easy and fast selection process due to taxonomies the project characteristics, it is better for selection of different elicitation techniques for complex systems which have different project domains.

6. Conclusion and Future Work

The conclusions drawn from the case study and the functionality provided by the model suggested that the initial objective of the research "develop knowledge based selection of appropriate requirements elicitation techniques based on taxonomy of the project types" has been achieved. But the quality and precision of the suggested model is dependent on the quality and number of parameters stored in the database. Therefore from this study it can be concluded that as the knowledgebase gets richer and richer, the quality and precision of the techniques suggested by the model will increase.

The main contribution of this research is that a new method is developed to select appropriate elicitation techniques based on taxonomy of the project type, and the following points can specifically be mentioned in this regard.

- Existing knowledge based system processes are classified and organized into different software project types. It is built with a clear structure with the intention of helping to select the most suitable techniques easily for the analyst.
- The case study and the prototype developed are strong indications that the taxonomy of the project characteristics is a valuable asset to extend the development of a system which can help selection of suitable elicitation techniques.
- The questionnaire analysis shows that only 8 elicitation techniques are used by local private software development companies. But our research added more elicitation techniques to increase the options of selecting elicitation techniques for the analyst or requirement engineer.

The intent of the proposed method is not solving all software engineering problems; instead, it provides a systematic way to develop elicitation techniques selection system. Given the merits, the research has some limitations and the following are the research directions for the future:

- Even through the overall structure of model is complete, more software domain knowledge would further increase the help provided by the model.
- There are limited software project attributes and elicitation techniques. Hence more project attributes and elicitation techniques increase the help provided by this model.
- The model is still a prototype and additional functionality has to be implemented. The two main tasks are to further improve the model and the completion of the system tool.
- In order to validate the effectiveness of the system model, further case studies using diverse software in the health domain knowledge are necessary.
- In the suitability techniques assessment process of the model, the project attributes vs.

the elicitation techniques data need further refinement.

References

[1] Wondwossen Elssa, "Integrated Approach for Software Project Management Methodology Selection", Unpublished Master's Thesis, HiLCoE, Addis Ababa, 2012.

[2] Q. Zhang and A. Eberlein, "Architectural Design of Intelligent Requirements Engineering Tool", Canadian Conference on Electrical and Computer Engineering, CECE'03, Montreal, QC, 2003, pp. 1375.

[3] Semahegn Abebe, "A Framework for Requirements Elicitation Selection", Unpublished Master's Thesis, Addis Ababa University, 2004.

[4] Didar Zowghi and Chad Coulin, "Requirements Elicitation: A Survey of Techniques, Approaches and Tools", Springer, Germany, 2005, pp. 19-46.

[5] Ann M. Hickey and Alan M. Davis, "Requirements Elicitation and Elicitation Technique Selection": A Model for Two Knowledge-Intensive Software Development Processes, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2003.

[6] Li Jiang, "A Framework for the Requirements Engineering Process", Unpublished PhD Dissertation, University of Calgary, 2005.

[7] Ann M. Hickey Alan M. Davis, "Elicitation Technique Selection: How Do Experts Do It?" Proceeding of the 11th IEEE International Requirements Engineering Conference.

[8] http://en.wikipedia.org/wiki/Software_ maintenance accessed on Aug 24, 2012.

[9] Kagan Erdil *et al*., "Software Maintenance as part of the Software Life Cycle", Department of Computer Science, Tufts University, December 16, 2003.

[10] Pekka Forselius, "Software Development Program Characteristics", Software Technology Transfer, Finland, 2011.

[11] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: a Roadmap", Proceeding of the Conference on the Future of Software Engineering, 2000.

[12] Tuure Tuunanen, "A New Perspective on Requirements Elicitation Methods", Journal of Information Technology Theory and Application 5 (3) 2003, pp. 45-62.

[13] Sai Ganesh. Gunda, "Requirements Engineering: Elicitation Techniques", Unpublished Master's Thesis, Software Engineering, Department of Tech, Math and Comp Science, 2008.

[14] Yannis Tzitzikas, *et al*., "Extended Faceted Taxonomies for Web Catalogs", Department of Computer Science, University of Crete, Greece, ICS-FORTH, 2002.

[15] Yannis Tzitzikas, *et al*., "An Algebraic Approach for Specifying Compound Terms in Facted Taxonomy", Istituto di Scienza e Tecnologie dell' Informazione, CNR-ISTI, Italy, 2003.

[16] Bhargav Kanagal, *et al*., "Supercharging Recommender Systems using Taxonomies for Learning User Purchase Behavior", The xth International Conference on Very Large Data Bases. *Proceedings of the VLDB Endowment,* Vol. X, No. Y, 2012.

[17] William B. Frakes and Ricardo Baeza-Yates, "Information Retrieval: Data Structures & Algorithms", 2004.

[18] Sujatha R, Bandaru Rama Krishna Rao, "Taxonomy construction techniques – issues and challenges", Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2, No. 5 Oct-Nov 2011.