

Domain Knowledge Based Requirements Change Management Framework: Case of Airlines Systems

Almaz Tsegaye

Ethiopian Airlines, Addis Ababa, Ethiopia
almazt96@gmail.com

Mesfin Kifle

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

Abstract

Requirements change management is becoming a challenging task since changes evolve frequently due to the dynamic situations in business and governance. Changes in business and governance affect the business context and the way systems manage. Traditionally, business and software system experts work together to analyze the impact of change on the existing systems. In other words, the change enforces experts to maintain their existing domain knowledge.

In this paper, a domain knowledge based framework is intended to capture the basic knowledge as well as the accumulated experiences of an enterprise to facilitate requirements change management. During the research, a survey was conducted and analysis was done to get a clue to solve the research questions such as “What kind of framework is appropriate to represent existing Domain Knowledge and for Requirement Change Management?” and “How do we design and validate the framework?”

A framework for requirement change management is proposed for the case of Airline Systems. Based on knowledge types, their properties and relationships, the impact analysis algorithm is formulated. In order to validate and verify the framework, a case study was conducted on Airline Revenue Accounting and a prototype was developed for framework verification.

Keywords: Domain Knowledge; Domain Knowledge Representation; Framework; Requirements Change Management; Requirement Management

1. Introduction

The primary measure of success of a software system is the level to which it meets the purpose for which it was projected, and RE (Requirements Engineering) is the process of discovering that purpose. The RE process commonly includes five phases: Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Documentation, Requirements Verification and Validation, and Requirements Management [1]. As presented in [1], the exposition of a set of assumptions, concepts, values, and practices that constitutes a way of understanding the research within a body of knowledge is called a framework.

Domain Knowledge change is inevitable in Airlines industry due to various factors. The research objective was to study the existing Requirements Change Management process, reviewing concepts about Domain Knowledge base Requirement Change

Management, identifying Domain Knowledge that can be applied for Requirement Change management, analyzing, modeling and development of technique and tools to validate the framework.

In order to study the existing Requirements Change Management process; interview was conducted with Domain and IT Experts. To complement the findings obtained during interview, work flow observation and secondary sources of information, such as documents, policies, and manuals have been assessed. Finally, framework was proposed, prototype for framework verification was developed and case study on Airline Revenue Accounting System to verify the framework was done.

The remaining part of the paper is organized as follows: Section two briefly explains information related to the research. Section three deals with the proposed framework design and modeling. In Section four, experimentation and results are presented.

Section five deals with related work to put conceptual foundation, and finally Section six presents the conclusion.

2. Background

Requirement Management involves: controlling changes to the requirements baseline, keeping project plans current with the requirements, controlling versions of both individual requirements and requirements documents, tracking the status of the requirements in the baseline, and managing the logical links between individual requirements and other project work products [2].

Once the initial requirements for a body of work are in hand, we must cope with the inevitable changes that customers, managers, the development team, and others request. Effective change management demands a process for proposing changes and evaluating their potential cost and impact on the project. Requirements change is a reality for all software projects, but disciplined change-management practices can reduce the disruption that changes can cause [2].

As explained in [2], impact analysis is a key aspect of responsible requirement management. The analysis examines the proposed change to identify components that might have to be created, modified, or discarded and to estimate the effort associated with implementing the change. Before a developer says, "Sure, no problem" in response to a change request, s/he should spend a little time on impact analysis. It provides better understanding of the implications of a proposed change, which helps to make informed business decisions.

Therefore, in order to implement change impact analysis solution, domain knowledge based change management helps to easily trace change dependency and to make informed decision to accept or reject the change, based on change impact analysis. Change impact analysis is identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change [3].

Domain knowledge is a form of knowledge which is used to refer to the knowledge possessed by the

people about certain business processes in an organization [4]. The authors in [5] stated that domain knowledge is acknowledged to be important for the acquisition and validation of requirements specification. Unfortunately, requirements validation and acquisition is one of the most error prone and costly activities in system development which has led a number of researchers to investigate how the process of requirements capture and validation may be improved. Previous approaches have varied from development of powerful specification language, to investigation of declarative, rule based paradigms and embedding domain knowledge in CASE (Computer Aided System Engineering) tools to allow intelligent requirements validation. A possible solution to the Domain Knowledge acquisition dilemma is to use more generic Domain Knowledge.

3. Framework Design and Modeling

According to [6], framework is a hypothetical description of a complex entity or process. Framework is one of the possible ways for finding solutions to problems. Frameworks are commonly used to combine the techniques on a specific problem, topic or area [1].

In order to investigate the necessity of Domain Knowledge based Requirement Change Management Framework, a survey was conducted which intends to assess the current Requirement Change management process for Airlines systems. The survey result revealed the existence of requirement change management problem. This section presents the proposed Domain Knowledge based Requirement Change Management Framework and Domain Knowledge Model. The proposed framework consists of domain knowledge maintenance and change impact analysis. The elements of the Domain Knowledge are rule, processes or tasks and system. The proposed framework is depicted in Figure 1.

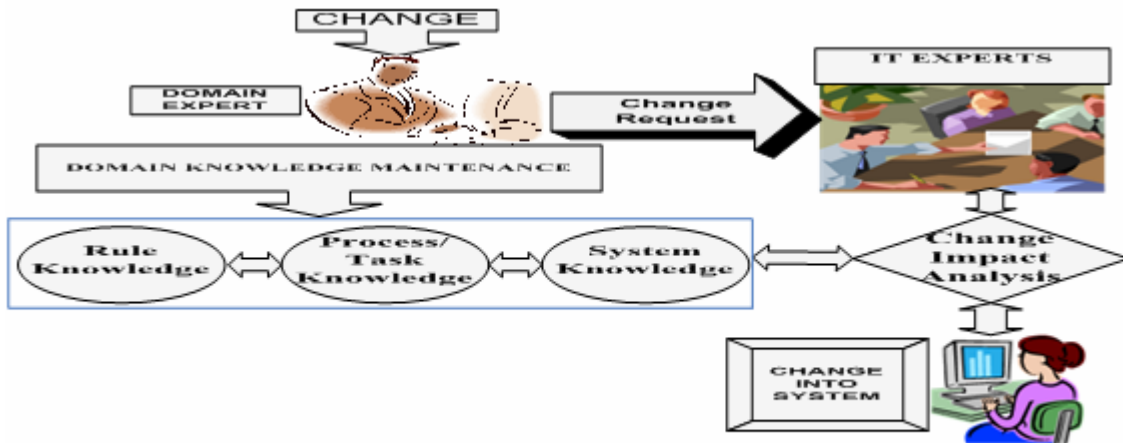


Figure 1: Domain Knowledge based Requirement Management Framework

When the change happens due to industry regulation change or any business area requirement, first the Domain Expert will do analysis work based on change source information. The analysis could be like studying the new change implication from the source of the requirement, checking the existence of new knowledge due to the change in order to add new knowledge into Domain Knowledge. The change may happen to already existing Domain Knowledge which may require removal or update of existing knowledge. The proposed framework contains change request submission, knowledge maintenance, and change impact analysis. Domain Knowledge or an update to existing knowledge will be decided by the domain expert and later the IT expert will involve on software change impact analysis and will prepare the impact analysis report for Management decision. If required, s/he can query the existing knowledge to check dependency of the new change on existing knowledge or previous requirement. The domain expert can do at high level the impact analysis of the change on the existing system. Then the domain expert can forward the requirement to the IT section.

The IT Expert further studies the impact and dependence of the new requirement in detail, but this time most of the task is already done by the Domain Expert. Hence, the IT expert can manage the change timely and with better quality. The IT Expert identifies Airline systems which will be impacted

due to regulation change or any change request. The new framework will also reduce rework on change request, because the domain expert will have a role to play in the new change request rather than just forwarding the request to the IT section. The Domain Experts will participate on impact analysis which will help them to understand the volume of work required by the IT section. It will also help for knowledge reuse and for other new software development.

The knowledge representation model of the system, tasks, processes, and rules are shown in Figure 2. The model is done using frame based knowledge representation. The model shows the dependency among four knowledge types. The tasks which exist in the Airline domain are highly dependent on rules and regulations of the organization itself and other regulatory bodies. For instance, all flight operations and ticket sales depend on each country's and regulatory bodies' rules. Accordingly, processes are also guided by rules indirectly, since the task(s) under each process are done based on rule. On the other hand, systems, subsystems and modules are highly interrelated; one is the subset of the other. In addition, a process depends on the system, tasks and modules have relations as they are subsets of a system.

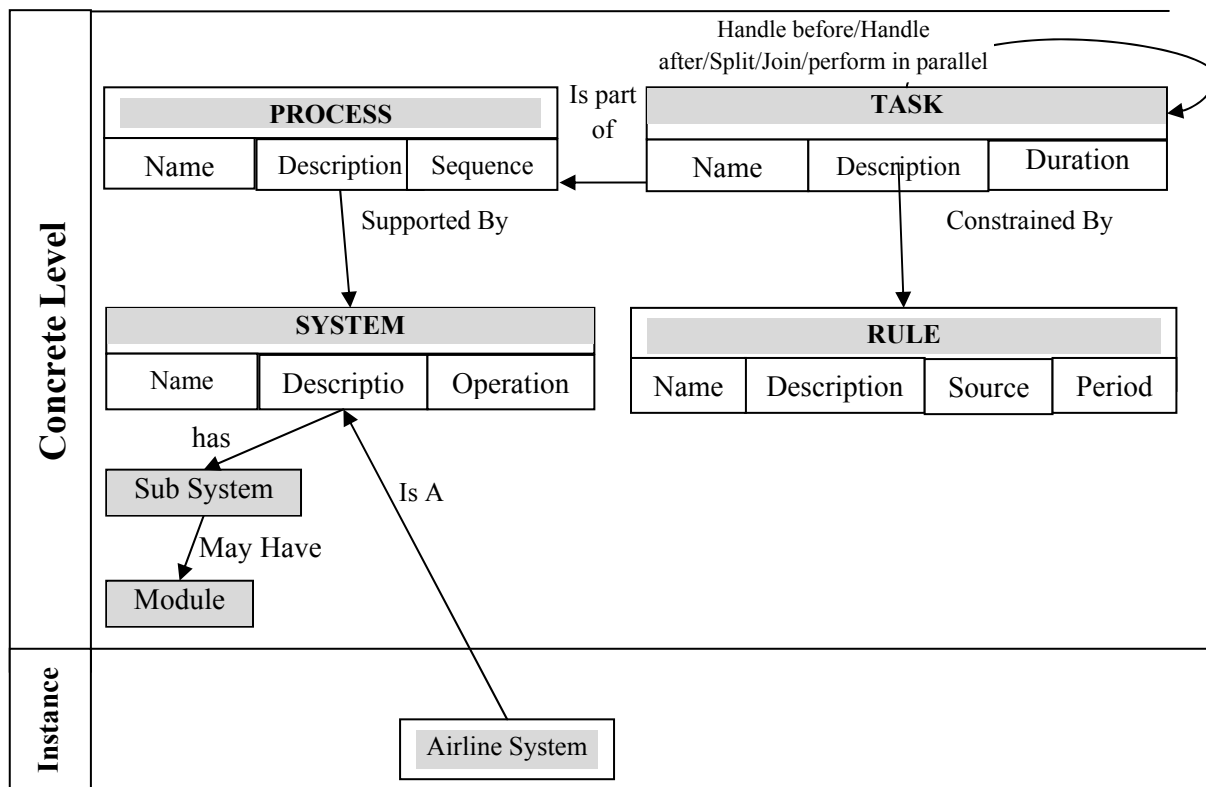


Figure 2: Domain Knowledge Representation Model

4. Experimentation and Results

The aim of the experimentation is to demonstrate the output of the research and to verify and validate the proposed framework. In order to verify our framework with practical example, Airlines Revenue Accounting was chosen as a case study. The reason we choose Revenue Accounting was because it encompasses all the knowledge types (process, task, system and rule/regulatory bodies and company rules) which are in the framework. As per the researchers' knowledge, there are a lot of new change request and requirement change due to various reasons.

To conduct the case study, it is necessary to have a system to test the framework and the model. A Domain Knowledge and change request repository RDBMS (Relational Database Management System) was chosen, which is Microsoft Access. The programming language used for implementation was C#. The architecture of the prototype has a client side and at backend database for data storage.

In the framework, the Domain Knowledge is classified into three main areas. They are rule, process or task, and system knowledge. The technique used to utilize the Domain Knowledge was

by storing the knowledge in a database to share and apply for change impact analysis.

The prototype designed had a capability of change request and domain knowledge maintenance and the change impact analysis report extraction. There are user interfaces which have a capability of record maintenance for change request, rule, task, process, system, and module. In addition to the data maintenance operations, it has the capability of searching the related or impacted knowledge from process, task, rule and system tables. For the case study sample, previous change request data has been taken. When the change occurs, the domain expert searches for change request record to check if the change already exists by the search request command using as a parameter change request keyword. The keyword is more related to domain knowledge classification or terms in the domain like, for example, the request if it is related ticket sales data entry system or module. The domain expert may use as a keyword like "Ticket" and search any request which exists previously by this same keyword. If it exists, the domain expert will ignore the data entry for change request. But s/he can check the status of the request whether it is active or not. If it is active,

s/he will follow up the status with his/her section or in the IT section. The actual status of the request may be waiting to change request approval, or it is being developed in the IT section. If the request is new, s/he will add the request into the system, search for change impact and the request is next handled by an IT expert after proper change request submission to the IT section in the form of a letter or e-mail. Then the IT experts will do additional change impact analysis using impact search command which exists on the user interface.

The user interface will handle change request and knowledge insertion, deletion, update and search. The screen could be used both by Domain Experts and IT experts to extract impact analysis. For the IT expert, the same user interface could be used for change impact analysis.

5. Related Work

The authors in [7] designed a framework and architecture supporting the utilization of Domain Knowledge embedded in software. The framework is based on identifying Domain Knowledge acquired during the development process (especially in requirements analysis) and formalizing it. They have identified seven fundamental concepts commonly used in modeling application domains: Actors, Roles, Resources, Services, Goals, Constraints, and States (Consisting of Attributes). They verified their framework by implementing a system and by classifying the system architecture into two parts: one represents the Domain Knowledge and the other is responsible for the actual processing (using this knowledge). Their work focused on explicitly extracting Domain Knowledge which is embedded within the system. It does not include the additional Domain Knowledge which exists external to the system like within Domain Expert's mind and in the various documents. The research work could be used to utilize part of the Domain Knowledge and the knowledge which exists within the system. Addressing the entire domain knowledge representation is advisable if a new requirement is requested which we can use the existing knowledge for requirement management. If we identify and represent the Domain Knowledge right from the origin, it will be easy to utilize during requirements

management. The use of Domain Knowledge which exists in software itself may not be enough for Requirements change management. The entire Domain Knowledge could not be found in existing software. Our research recommendation is to represent Domain knowledge which exists within the mind of Domain Experts, in policies and procedures, and systems by storing in separation to the system itself so that knowledge maintenance can be easy.

The research work in [8] contributed a method and framework for Domain Knowledge assisted requirements evolution. Their platform facilitates requirements elicitation for a team which are geographically in different areas but are working as a team for the same project. This work can be referred to as a baseline that can be applied in requirements change management process perspective. What the researchers learned considering domain knowledge as "seeds" of requirements can potentially be a new viewpoint in Requirements Engineering. Their framework design and the way they apply the domain knowledge to fulfill requirement elicitation parameters like completeness, consistency, ambiguity, and correctness is well taken. But the objective of our research work was to utilize domain knowledge as a means to extract change impact analysis.

The framework in [9] helps to integrate requirements management and design knowledge reuse. The framework is then demonstrated using a case study example: vacuum pump design. Requirements are presented as a component of the integrated design knowledge framework. The proposed framework enables the application of requirements management as a dynamic process, including capture, analysis and recording of requirements. It takes account of the evolving requirements and the dynamic nature of the interaction between requirements and product structure through the various stages of product development.

6. Conclusion

Recently, the business environment is dynamically changing which in turn affects existing software systems. As a result, the business

knowledge and system requirements will change. Requirements change management is focused to conduct change impact analysis. In order to study this problem, a survey was conducted which intends to assess the current Requirement Change management process for Airlines systems. The survey result revealed the existence of requirement change management problem. Therefore, in order to solve the existing Requirement Change Management problem, this paper proposed a Domain Knowledge based Requirement Change Management Framework. The proposed framework consists of domain knowledge maintenance and change impact analysis. Experimentation is done to demonstrate the output of the research, to verify and validate the proposed framework. In order to verify the framework with practical example, Airlines Revenue Accounting Domain was chosen as a case study.

References

- [1] M. Husnain, M. Waseem, and S. A. K. Ghayyur, "An Interrogative Review of Requirement Engineering Frameworks," *International Journal of Reviews in Computing*, 2009.
- [2] Karl E. Wiegers, *Software Requirements, Second Edition*. 2003, pp. 516.
- [3] "Change Impact Analysis," 2012. http://en.wikipedia.org/wiki/Change_impact_analysis. [Accessed: 10-Mar-2012].
- [4] "Domain Knowledge." <http://www.freetechexams.com/computers-tips/computer-tips/domain-knowledge.html>. [Accessed: Aug-2011].
- [5] A. G. Sutcliffe and N. A. M. Maiden, "Use of Domain Knowledge for Requirements Validation," in *Proceedings of IFIP WG8.1 Conference on information System Development Process, Como Italy, 1993*, no. September, pp. 1-3.
- [6] "Freedictionary." <http://www.thefreedictionary.com/framework>. [Accessed: 26-Apr-2012].
- [7] E. Rubin and Y. Wand, "A Framework Supporting the Utilization of Domain Knowledge Embedded in Software," *Reproduction*, vol. 83, 2007.
- [8] P. Rose, M. Kumar, N. Ajmeri, M. Agrawal, V. Sivakumar, and S. Ghaisas, "A Method and Framework for Domain Knowledge Assisted Requirements Evolution (K-RE)," in *Proceedings of CONSEG-09: International Conference on Software Engineering Dec 17-19, Chennai, India*, pp. 87-97.
- [9] D. Baxter and et al., "A framework to integrate design knowledge reuse and requirements management in engineering design," *Robotics and Computer Integrated Manufacturing Systems*, pp. 1-9, 2007.