2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)
Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore

Design and Development of Ontology for Risk Management in Software Project Management

C.R.Rene Robin [1 +] and G.V. Uma [2+]

[1] Research Scholar, Dept. of CSE, Anna University Chennai, Chennai-25
[2] Asst. Professor, Dept. of CSE, Anna University Chennai, Chennai-25

Abstract. One of the definitions of risk management is "the identification of the hazards and possible problems, the evaluation of their importance and the drawing up of plans to monitor and deal with those problems". Failure of some sort has been a common occurrence in the software development milieu. The survey indicates that more than 25% of all software development projects are cancelled outright before completion and something like 80% overrun their budgets. This paper describes a framework that can be used to document the knowledge gained through text book, domain experts and personal experience. And it describes a method for facilitating the systematic and repeatable identification of risks associated with of a software-dependent development project.

Keywords: Risk Management, Software Project Management, Taxonomy, Ontology, Protégé, etc.

1. Introduction

A lack of project management skills could affect many different activities at many different stages of the project. For example, the risk 'lack of user commitment led to delays in acceptance testing' clearly relates to one specific stage in a project. A typical approach to risk management is to maintain a risk register during a project. As the project progresses, some risks will cease to have an impact, while the threat of others may grow. It has been found helpful to produce a template separated into the generic stages of the development life cycle and to locate the occurrence of risk factors and their outcomes within these physical stages, as well as linking the risk factors through chains of cause and effect. There can be no doubt that risk management is an important activity in the software engineering [1] area. The literature on risk management for software maintenance is much scarcer. On the other hand, software maintenance projects do present specificities that imply they offer different risks than development. This suggests that maintenance projects could greatly benefit from better risk management tools. One step in this direction would be to help identifying potential risk factors at the beginning of a maintenance project. For this, we propose ontology of possible risks for software management projects. The ontology was created from an extensive survey of risk management literature, to list known factors for software development and an extensive survey of maintenance literature, to list known problems that may occur during maintenance.

2. Proposed Work

A lack of project management skills could affect many different activities at many different stages of the project. For example, the risk 'lack of user commitment led to delays in acceptance testing' clearly relates to one specific stage in a project. A typical approach to risk management is to maintain a risk register during a project. As the project progresses, some risks will cease to have an impact, while the threat of others may grow. It has been found helpful to produce a template separated into the generic stages of the development

[+] Corresponding author. Tel.: + (91-9840222341);
E-mail address: (crrenerobin@gmail.com).

life cycle and to locate the oc currence of risk factors and their outcomes within these physical stages, as well as linking the risk factors through chains of cause and effect. There can be no doubt that risk management is an important activity in the software engineering area. The literature on risk management for software [3]maintenance is much scarcer. On the other hand, software maintenance projects do present specificities that imply they offer different risks than development. This suggests that maintenance projects could greatly benefit from better risk management tools. One step in this direction would be to help identifying potential risk factors at the beginning of a maintenance project. For this, we propose ontology of possible risks for software management projects. The ontology was created from an extensive survey of risk management literature, to list known factors for software development and an extensive survey of maintenance literature, to list known problems that may occur during maintenance.

3. Ontology

Ontology describes the concepts and relationships that are important in a particular domain, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary. Ontologies range from taxonomies and classifications, database schemas, to fully acclimatized theories. In recent years, ontologies have been adopted in many business and scientific communities as a way to share, reuse and process domain knowledge [3]. Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services.

4. Protégé

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies[5]. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. The Protégé-Frames editor enables users to build and populate ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). In this model, ontology consists of a set of classes organized in a subsumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties. We have used this method for constructing risk ontology. And constructed a class hierarchy of various possible software risks that may occur in a software project.

5. Methodology Used

The ontology of software development risks maps the characteristics of software development and hence of software development risks. The method described in this paper presents a disciplined and systematic way to identify risk in a software-dependent system development. This method allows risks to be identified without justification and without a proposed solution. The risk identification method surfaces and clarifies the uncertainties and concerns of a project’s technical and managerial staff.

6. Risk Ontology

Central to the risk identification method is the software development Ontology[2]. The ontology provides a framework for organizing and studying the breadth of software development issues. Hence, it serves as the basis for eliciting and organizing the full breadth of software development risks—both technical and non-technical. The ontology also provides a consistent framework for the development of other risk management methods and activities.

7. Taxonomy Developed from the Identified Risks

This is the prior step of ontology construction. The following is the part of taxonomy developed manually from the identified risks involved in the software development process.

1. Product Engineering
    - 1.1. Requirements
        - 1.1.1. Stability
        - 1.1.2. Completeness
        - 1.1.3. Clarity
        - 1.1.4. Validity
        - 1.1.5. Feasibility
        - 1.1.6. Precedent
        - 1.1.7. Scale
    - 1.2. Design
        - 1.2.1. Functionality
        - 1.2.2. Difficulty
        - 1.2.3. Interfaces
        - 1.2.4. Performance
        - 1.2.5. Testability
        - 1.2.6. H/W Constraints
        - 1.2.7. Non-Developmental S/w
    - 1.3. Engg. Specialties
        - 1.3.1. Maintainability
        - 1.3.2. Reliability
        - 1.3.3. Safety
        - 1.3.4. Security
        - 1.3.5. Human Factors
        - 1.3.6. Specifications
2. Development Environment
    - 2.1. Development Process
        - 2.1.1. Formality
        - 2.1.2. Suitability
        - 2.1.3. Process Control
        - 2.1.4. Familiarity
    - 2.2. Management Methods
        - 2.2.1. Monitoring
        - 2.2.2. Personnel
        - 2.2.3. Quality Assurance
    - 2.3. Work Environment
        - 2.3.1. Quality Attitude
        - 2.3.2. Cooperation
        - 2.3.3. Communication
        - 2.3.4. Morale
3. Program Constraints
    - 3.1. Resources
        - 3.1.1. Schedule
        - 3.1.2. Staff
        - 3.1.3. Budget
        - 3.1.4. Facilities
    - 3.2. Contract
        - 3.1.1. Type of Contract
        - 3.1.2. Restrictions
        - 3.1.3. Dependencies

Fig. 1: Taxonomy developed from the identified Risks

8. Implementation Results

8.1. Creation of Class and Subclass

Classes represent concepts in the domain and not the words that denote these concepts. Here top down development process is handled which starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts. The classes of ontology may be extensional or intentional in nature. A class can subsume or be subsumed by other classes. In the Fig.1 we can view the entire class hierarchy. The parent class and the child classes are clearly visible.

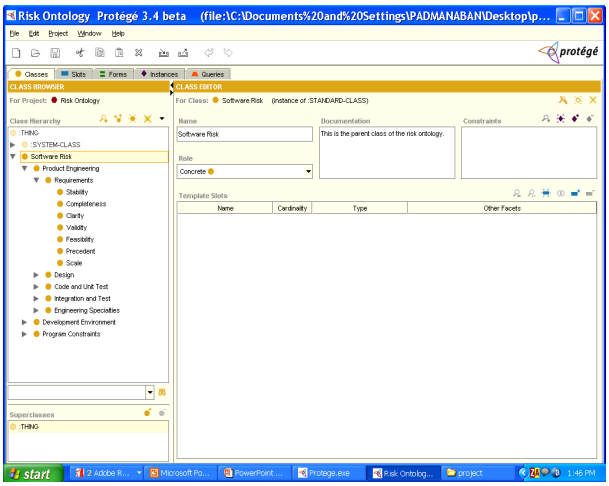

Fig. 2: Class Editor showing Class Hierarchy

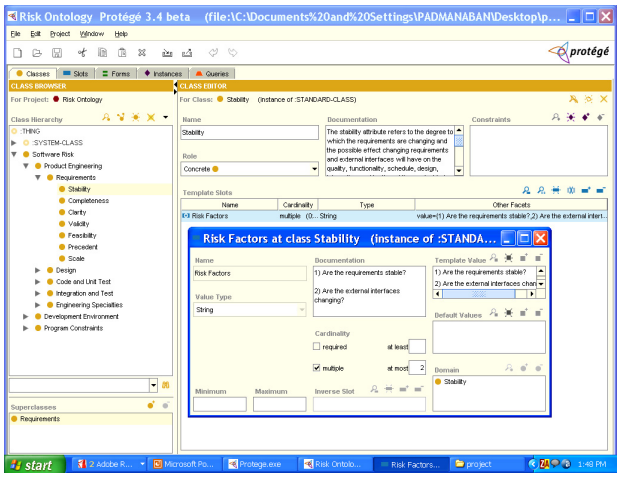

Fig. 3: Protégé Slot Editor

8.2. Creation of Instances

All subclasses of a class inherit the slot of that class. In the fig.3 we can see the protégé slot editor. We can create one or more slots for each class. The slots can have either a single value or multiple values.

8.3. Parent Class

Fig.4 shows the parent window. This is used to view the parent for the selected class. The user has selected the class completeness. The parent class for completeness is requirements. It is used to determine whether the given requirements are complete or not.

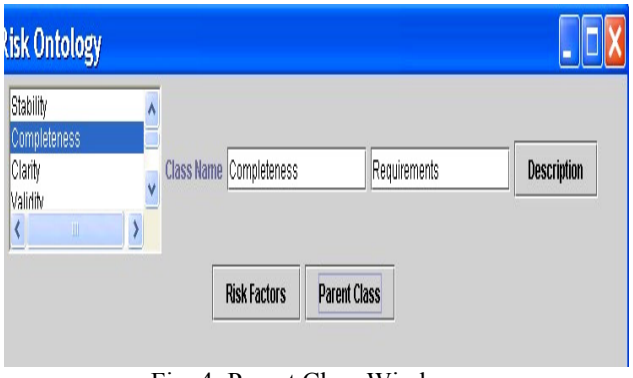

Fig. 4: Parent Class Window

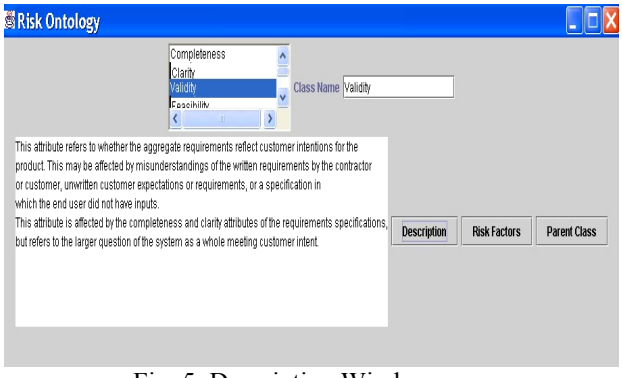

Fig. 5: Description Window

8.4. Description Window

Fig.5 shows the description window. The user can select the class and if he wishes to know more about the selected class then he can click the description button. The below screen shot shows the description about the class validity. The class validity is highlighted in the list box.

8.5. Risk Factors

Fig.6 is called risk factor window and is used to view the associated risk factors of the selected class. The risk factors of the selected class are indicated below. The risk factors listed below are by no means complete in any aspect. It is only a small indication about the possible risks associated with that particular class.

8.6. Parent Class Interface

The Fig.7 shows the user interface of the get child class. The parent classes are listed in the list box. The user has to select a parent class from the given list. Then he can get a list of all the child classes for the

selected parent class. This will facilitates to know what the subclasses are and also to observe the pattern of super-sub class hierarchy.

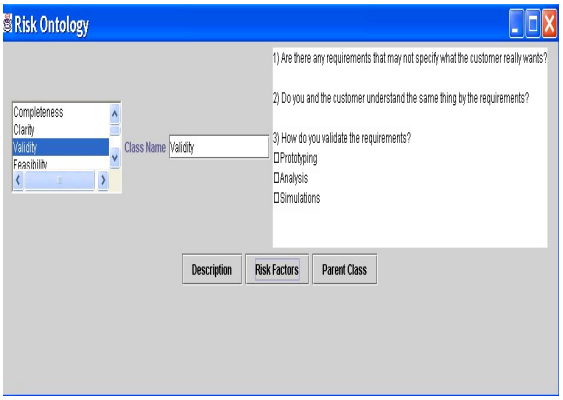

Fig. 6: Risk Factor Window

Fig. 7: Parent Class User Interface

9. Conclusion and Future Enhancement

In this paper we have constructed risk ontology for the domain of software project management. The constructed Ontology can offer far more information than purely textual documentation formats and in a compact and highly-visual format. The framework described in this paper can be used to document the knowledge that team members gained through experience. The elicitation process involves the identification of risk factors, their likelihood of occurrence, and their likely impact on other risks within the cause and effect relationships at the heart of the model. The use of ontology throughout this process has been observed to facilitate 'brainstorming' and to achieve consensus in a more intuitive and effective way than with more conventional approaches.

We have constructed risk ontology for software project management. This is general purpose ontology and it is not constructed for any specialized project. It is constructed in the domain of software engineering and it is not applicable to any other domain. We can extend this ontology for specialized domain or it can be changed into more general purpose risk ontology. The size of the constructed ontology can be increased by adding more number of candidate risk factors. The information that is stored in the ontology can be retrieved by using queries. This work can also be enhanced by developing additional methods not only for risk identification but also for risk analysis, planning, tracking, and control, results of the field tests encourage and claim that the described method is useful, usable, and efficient.

10. References

[1] Dillon, T.S., Chang, E., Wongthongtham.P, 'Ontology-Based Software Engineering – Software Engineering 2.0',19th Australian Conference on Software Engineering, 2008, 26-28 March 2008, Page(s):13-23

[2] Pornpit Wongthongtham, Elizabeth Chang, Tharam Dillon, and Ian Sommerville 'Development of a Software Engineering Ontology for Multisite Software Deveopment'. IEEE Transactions on Knowledge and Data Engineering, Vol.21, 2009

[3] Jawad Makki, Anne-Marie Alquier, Violaine Prince, 'An NLP-based ontology population for a risk management generic structure', 5th International Conference on Soft Computing as transdisciplinary Science and Technology 2008, pages 250-355.

[4] Christian Cuske, Tilo Dickopp, Stefan Seedorf 'An ontology based platform for knowledge based simulation modeling in financial risk management', European simulation and Modeling Conference 2005

[5] Ricardo de Almeida Falbo, Ana Candida Cruz Natali (2006), 'Ontology based software development environment'.

[6] www.stanforduniv.com