Predicting Requirements Changes by Focusing on the Social Relations

Takako Nakatani[1] **Toshihiko Tsumaki**[2]

[1] Graduate School of Business Sciences, University of Tsukuba,
3-29-1, Ohtsuka, Bunkyo-ku, Tokyo, 112-0012, Japan
Email: nakatani@gssm.otsuka.tsukuba.ac.jp

[2] Information Systems Architecture Research, National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
Email: tatsumaki@nii.ac.jp

Abstract

Software requirements are changed by various factors. Stakeholders that are analysed in traditional requirements engineering are mainly requesters or decision makers with regard to the requirements specifications. Such stakeholders are selected as intentional actors in the i* framework. The novel point of this paper is that we focus on the world of parties who are the environmental factors of the requirements of the intentional actors. Our purpose is to propose a method to predict requirements changes by focusing on the social relations. In this paper, we present our method and evaluate its effectiveness through a simulation of requirements changes.

Keywords: requirements engineering, requirements changes, requirements volatility, stakeholder analysis, social relations

1 Introduction

Requirements volatility produces various negative effects on the software development process: reduction of performance (Zowghi & Nurmuliani 2002) and increase of costs (Nurmuliani et al. 2004) to the project. How to cope with requirements volatility is one of significant themes in requirements engineering. If we can predict requirements volatility, our design techniques work well to prevent the deterioration of software. Design patterns (Gamma et al. 1995) is one of the solutions.

Basically, requirements are volatile. According to the observation of requirements elicitation (Nakatani et al. 2008), not a small number of requirements are elicited in the middle or late stages of projects. There have been researches done on requirements volatility. Ebert and Man focused on the problems that cause requirements volatility (Ebert & D. 2005). Also, the risks involved in requirements volatility have been discussed by Williams et al (Williams et al. 2006). According to our closed discussion with practitioners, the environmental factors of stakeholders have not been analysed well. The motivation of our study is to develop a method to analyse the environments surrounding stakeholders who are involved in the requirements analysis. What kinds of environmental factors affect these stakeholders?

The purpose of this paper is to predict requirements changes through the analysisation of the environmental factors that may or may not have an impact on the requirements. There are multiple parties in the environment. For example, executives, competitors, cooperative organisations, the natural environment, etc. We focus on these parties and the "social relations" between parties. Changes in social relations sometimes force the stakeholders to change their requirements. If a requirements analyst only analyses the requirements of stakeholders, the changes in social relations are set a side, and thus, out of the analyst's scope. As a result, the analyst cannot realise requirements changes that may be caused from the social relations. We know that the analysts and stakeholders may not be able to control or manage such changes, but we have confidence that we can predict the changes that cause volatility within the requirements. In this paper, we present the effectiveness from the perspective of social relations in order to predict requirements volatility. It will help us clarify one of the aspects of requirements changes.

The structure of this paper is as follows: Section 2 presents the perspective of social relations to predict requirements changes and define the social relations; In Section 3, we describe a method to analyse these relations. After describing the method, we apply the method to an example in order to clarify the effectiveness of the method to predict requirements changes caused by the social relations in Section 4. More precisely, we will also show a conceptual model of the example domain, and designate the roles of conceptual analysis and social relations. Before we conclude the paper in the final section, we discuss the related work in Section 5.

2 Social relations

In this section, we will outline the strategic dependency model of i* (Yu 1997). Then, we introduce social relations that affect requirements volatility.

2.1 i* framework

The i* framework consists of two models: the strategic dependency model (SD model) and the strategic rationale model (SR model). In this paper, we focus on the SD model, and moreover, we extend the model with social relations.

The SD model contains dependency relations between intentional actors within the analysing world. The intentional actors are the stakeholders of the developing system. There are four dependency relations between actors: task, resource, goal, and soft goal. These dependency relation types are represented by a hexagon, rectangle, oval, and cloud, respectively in the SD model. With each relation, the symbol "D" is used, which represents the direction from a dependee to a depender. Figure 1 is an example of the SD model. It covers a part of the stakeholders of a train service.

A requirements analyst can apply i* in order to extract the goals of each actor, then he/she analyses the goal-oriented analysis with the SR model. The i* framework is helpful to analyse the "why" aspect of requirements. The actors in the scope of i* have intention in deciding the requirements of the developing system. Hence, the environmental factors that affect the intention of the actors were set outside the scope of i*.

The environmental factors cause requirements changes. Though there are various factors, we regard the principles of these factors to be but a few. We have challenged ourselves to clarify the basic concepts of the environmental factors, and further, set them inside the scope of analysis in order to predict requirements changes.

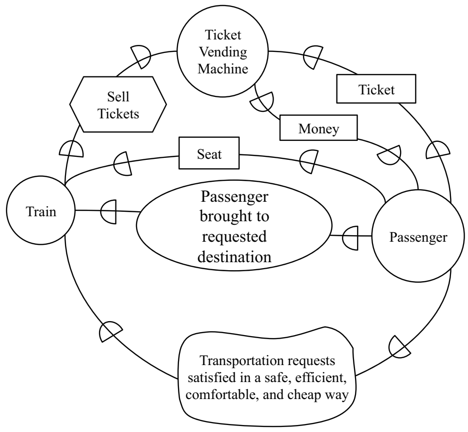

Figure 1: SD model: an example of train services.

2.2 Environmental factors

First of all, we divide the environmental factors that cause requirements volatility into internal and external factors of the software development project. Internal factors relate to the maturation of the project, and should be managed within the project. Examples of internal factors are, knowledge of engineers to understand the requirements correctly, technical issues on requirements analysis and design, and maturity of management. Conversely, external factors are uncontrollable for the project. These external factors include economic liberalism, changes in and of markets, changes in personnel, policy changes of organisations, and expansion in and of users' variation. The external factors are the targets of this paper.

In order to analyse external factors, we focus on their two properties: variability and fluidity.

- Variability
  As much as the domain becomes more complex, and the number of concepts increases, we can apply a generalisation-specialisation structure to the domain, and thus, resolve its complexity. When we introduce a more abstract concept from the similar concepts, then we are able to predict the addition of other similar concepts. The variability of the concept can be modelled with a conceptual analysis.
- Fluidity
  There is a kind of change that propagates from one party to another and further, causes other changes. We refer to the change as fluidity. The fluidity may induce crucial requirements changes.

We can, for example, visualise and analyse the variability of environmental factors with class diagrams of UML. In order to analyse the propagation of the fluidity, we extend the scope of the SD model with four types of relations between parties. In the next subsection, we define these four types of the relations.

2.3 Social relations

The fluidity arises within a relation between parties, i.e. the addition of a new party, the transformation of relations, and the deletion of relations. The volatility of the requirements volatility can be investigated through the fluidity of relations. We refer to a relation between parties as a social relation. Fiske classified human relations into four elementary forms: communal sharing, authority ranking, equality matching, and market pricing (Fiske 1992). We adopt them to four social relations: sharing relation, ranking relation, exchanging relation, and contracting relation. Then, the SD model is enhanced in order to analyse the environment of intentional actors by the four social relations. The media connected to the relations are, for example, gifts, offerings, sharing properties, strategies, rules, constraints, force, rights, etc. The detailed definitions of social relations are as follows.

- Sharing relation
  Sharing relations are relations between parties who share the common interests or cultures and feel that good things for one are good things for another. They are sometimes competitors or rivals. Characteristics of sharing relations are as follows.
  - Parties are connected by a sharing medium that brings common profit and property.
  - The action of parties is performed based on a common aim.
  - The parties sometimes try to obtain more shares than other parties. In order to solve such a competitive situation, the relation is transformed into another type.
- Ranking relation
  The upper ranking party has privileges over the lower ranking party. Ranking is introduced into societies such as militaries and corporations with social responsibilities. Characteristics of ranking relations are as follows:
  - The upper rankings receive payments from the lower rankings as a privilege of their authority, and in return, the lower rankings receive rewards from the upper rankings.
  - The lower party behaves according to the intentions of the higher party.
  - The ranking relation exists based on a power balance. If the balance is broken, the relation is transformed into another type.
- Exchanging relation
  This type of relation guarantees interdependence and fair exchange. Characteristics of exchanging relations are as follows.

- The same valued medium is exchanged between the members based on mutual agreement.
- If the medium is changed, the agreement may be broken, and then, the relation is transformed into another type.

• Contracting relation
Social contracts include tradition, rules, promises, etc. A party who breaks the contracts receives social punishments. The contracting relation is a basis of our modern social life. Characteristics of contracting relations are as follows.

- The value of medium is defined by authorised parties.
- The parties behave according to the social system.
- If the contents of the contract is changed, the relationship may be transformed.

3 Overview of the method

The method assumes that the SD model has been presented. The main part of our method is to provide a process of extracting four types of relations between parties. The relations are extracted from an open space. The following process is repeated until the effect on the intentional actors by a newly extracted relation is considered enough small to be ignored within the world of the SD model.

1. Focus on each intentional actor in the SD model.
2. Apply four types of social relations to each actor.
3. Define a medium on each relation that affects the intention of the actor. If no medium is found on the relation, delete the relation.
4. Define a party that is connected to the relation. Those parties may already be defined within the world of anaysis, and add the relation to the model.
5. Analyse the probability of transformation of each medium based on the knowledge and/or social experiences of the analyst.
6. Change the medium
There are two types of changes of a medium. The first type causes a transformation of the relation. This type of transformation changes the situation of the environment of the intentional actors, thus, the fluidity may change the intention of actors. The second type is a change of the medium without changing the type of the relation. This type can be modelled as the variability of a concept within a conceptual model.

The process is shown in Figure 2. In the next section, we show an example and simulate the process. According the simulation, we evaluate the effectiveness of the method.

4 Evaluation by example

4.1 Overview

We performed an empirical study on an application software development. The example was a rail transportation service support system of a passenger traffic company. The system was composed of functions

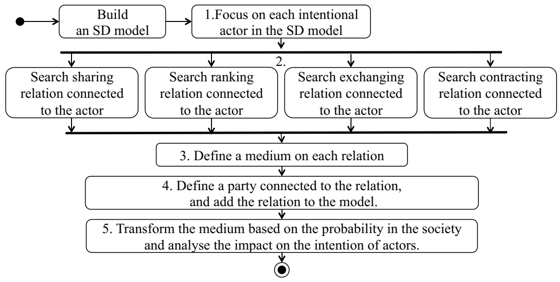

Figure 2: The process of the analysis.

to support safe and comfortable train services: train scheduling and rescheduling, traffic control, passenger reservation, ticket sales, users' claims to management, seat reservation, and so on. The main train route of *TrainCompany* had monopolised a certain district. *TrainCompany* told us that the purpose of the system is to support *a part of* the operations of the company.

The current requirements of the system are as follows:

- Req.1: Customers can reserve their desired seats through *several* different channels.
It is also possible to reserve seats for other days by using ticket vending machines in the station, by internet or telephone, and further, the tickets can be delivered through *various* channels.
- Req.2: Customers can cancel the reservation up to a specified time before the departure.
- Req.3: Customers can know the information of trains that they can take as well as the fare according to the recommended travel plan.
- Req.4: Customers can select the payment method: cash, credit card, or a prepaid card produced by the company.

We predicted the changes of requirements with the method. The results were evaluated by comparing them with the work of actual requirements changes.

4.2 Models in the application

We built two models in order to predict requirements changes. One is a conceptual model for analysing the variability of concepts, while the other is an extended SD model meant to analyse the fluidity of social relations.

4.2.1 Conceptual model

The conceptual model of the train service is shown in Figure 4. The coloured classes represent concepts outside the scope of the current system, but represent possible variablility of the concepts inside the scope. The variability is the source of requirements changes. For example, we know that the airplane service provides various classes of seats. Thus, new train services that provide such various kinds of reserved seats can be predicted. However, we need to analyse the fluidity of the environment with the extended SD model.

Simply, we can estimate the possibility of the modification of services through the conceptual model shown in Figure 4. HAZOP (IEC 2001) is a method to derive unexpected phenomena by using guide words, such as, over, less, slow, high, low, stop, clogged, intermittent, etc. For example, if some volume of oil goes in a pipe, such guide words as, over the volume, less volume, at a slower speed, at a higher speed, etc.,

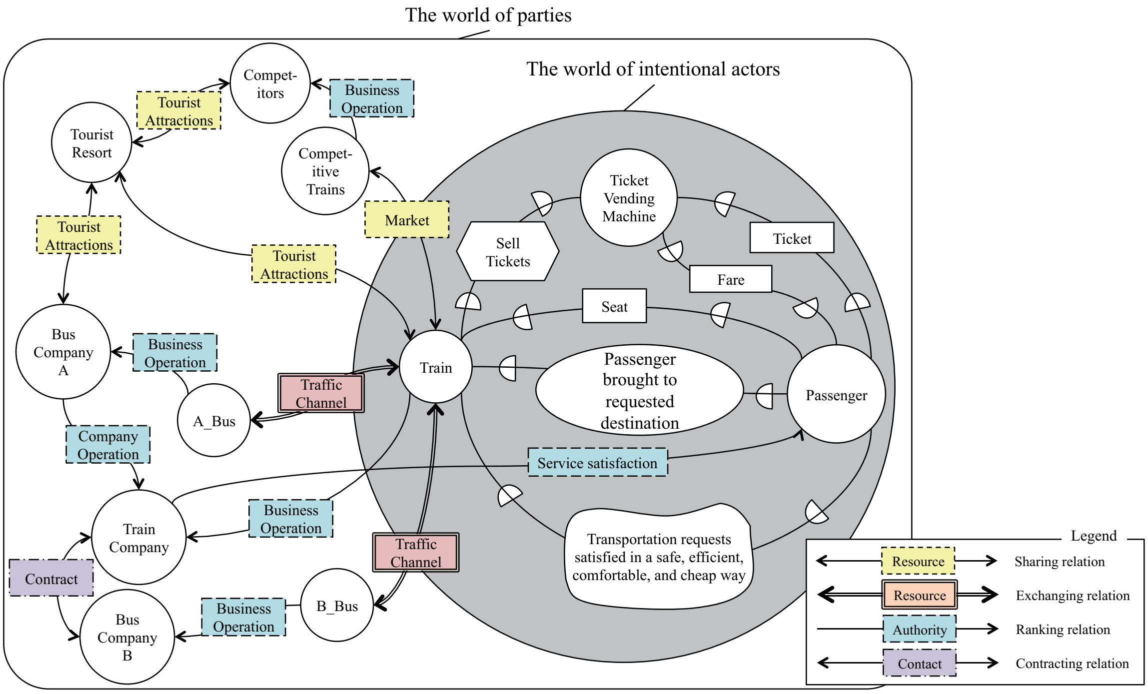

Figure 3: Extended SD model.

helps us define unexpected scenarios. We refer to the guide words of HAZOP and, define the following possible requirements changes. The attribute values and services can be a derived from a specialisation structure within the conceptual model.

Table 1: The possible change in ticket services.

Attribute	Current value	Possibility
from-to	fixed	flexible
transitPoint	fixed	flexible
fare	several kinds	emergency ticket
means of transportation	fixed	flexible
travel info.	none available	available
Service	**Current**	**Possibility**
Route variation	none available	available
Cooperation	none	air and bus

4.2.2 Extended SD model

The SD model was already shown in Figure 1. We extracted environmental factors with relations by focusing on *Train* within the SD model. After the third iteration of the analysis, we got the model shown in Figure 3. In the diagram, the scope of i* is shown as the greyed area.

In Figure 3, we show the legend for these four social relations. Basically, the arrows connect parties through objects. Only the ranking relation has a direction from a depender to a dependee. Other relations: sharing, exchanging, and contracting relation, are represented by bi-directional arrows. Especially, the double arrows of exchanging relation represent give and take processes. The extracted social relations are as follows.

- Sharing relations
  The following factors are connected with *Train* via sharing relations.
  - CompetitiveTrains
    CompetitiveTrains and *Train* share the market.
  - TouristResort
    TouristResort and *Train* share TouristAttractions.
- Ranking relations
  They are business operation, company operation, and service satisfaction. TrainCompany is a factor that is connected with *Train* via a ranking relation named business operation.
- Exchanging relations
  The following factors are connected with *Train* via exchanging relation. A_Bus and B_Bus have exchanging relations with Train via TrafficChannel.
- Contracting relations
  There is a contracting relation between TrainCompany and BusCompany_B.

4.3 Predicted requirements changes

Transformation of a social relation causes changes in the behaviour or roles of parties, and finally, they impact the intention of actors. We introduced four social relations and their characteristics. It is possible for those social relations to transform into other relations. By analysing such fragility, we can predict requirements changes with the conceptual model, and the extended SD model.

The following requirements on the services of Train are defined according to our analysis.

- ReqChange 1: Add new functions to operate new kinds of seats and establish various new fares.
- ReqChange 2: Introduce various fare options.
- ReqChange 3: Add new functions to manage new stations and operate flexible routes.

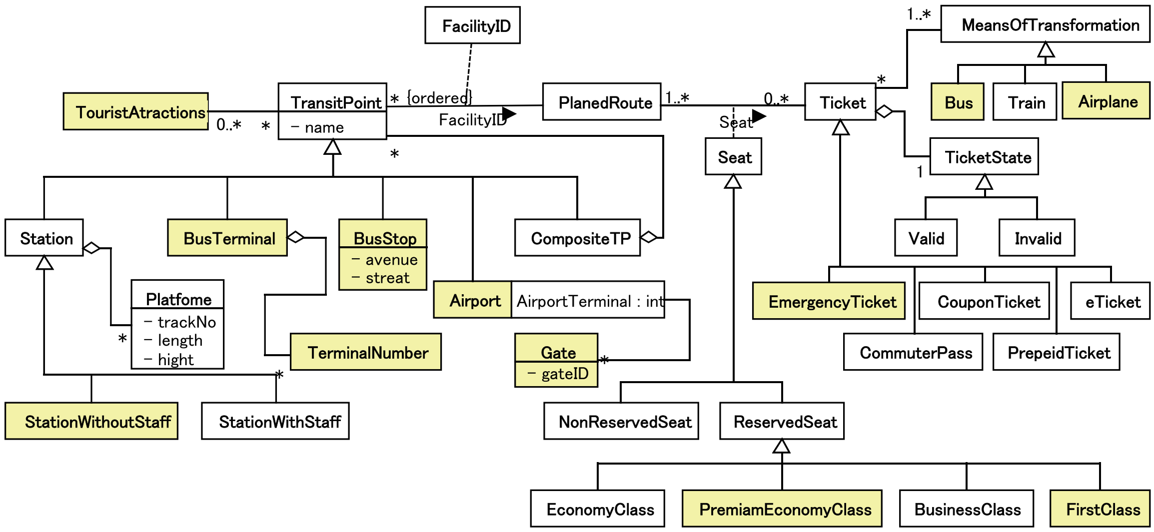

Figure 4: Conceptual model of the system domain

- ReqChange 4: Add new functions to handle the fares for the customers who transfer trains to/from a competitors' means of transportation.

We traced these requirements within the new system. *ReqChange 1* and *ReqChange 2* were derived from the conceptual model. *ReqChange 3* and *ReqChange 4* were predicted by analysing the extended SD model. *ReqChange 3* was found in the transformation of the sharing relation (labeled "TouristAttractions") to an exchanging relation (labeled "Station and/or route"). TouristResourt may require a new station for the Train, and TouristResourt invites passengers by Train. *ReqChange 4* was found in transformation of the sharing relation (labeled "Market") to a contracting relation (labeled "Inter-connection"). Here, we could predict the possibility of requirements changes with their rationales.

4.4 Discussion

In the real world, a competitor entered the area and started a faster train service. Then, TrainCompany decided to change the route to another route, of which, the time schedule was rather more sparse than the previous route. TrainCompany could out compete its competitor with regard to the speed. As a result, *ReqChange 3* and *ReqChange 4* were added. If we did not analyse the social relations, we could not realise the possibility of these requirements changes.

One of the changed requirements, which we could not discover, was a problem related to the location of reserved seats. Some of the stations on the new route were too short for the Trains, and so the protruding doors of the Trains must remain closed when the train stopped at the stations. Thus, before reserving seats for passengers, the system evaluates the length of all stations that the train stops at and, finds the available seats in the cars without the protruded doors. However, this additional requirement could not be predicted through our approach. The requirement depended on a business rule that states, TrainCompany has to provide a highly comfortable travel for passengers with seat reservations, and further, address the physical problem of the length of the stations. Our model does not have the ability to take the business rules into account. Our future work is to refine and evaluate the method by applying another example.

5 Related work

There are researches with regard to causes of requirements volatility. In our previous work, we introduced the concept of a speed of "requirements maturation" instead of requirements volatility (Nakatani et al. 2011). If requirements are volatile, their speed to mature is slow. The requirements maturation is calculated according to the stability of requirements and accessibility of stakeholders by a requirements analyst. Nurmuliani et al. categorised the causes of requirements volatility (Nurmuliani et al. 2004): they are evolution of customer and market needs, growth of users and developers and change of organisational policy. Bano et al. classified factors that cause requirements changes into three types: business, organisation and project, and estimated their impacts (Bano et al. 2012). However, they did not provide a method to analyse the environment factors.

According to the books of the late 20th century (MacAulay 1996, Sommerville & Sawyer 1997, Kotonya & Sommerville 1998), authors pointed out that the environmental factors cause requirements changes and focused on techniques to elicit requirements. Therefore, the environmental factors were not analysed in their scope. Alexander and Robertson developed the onion model to understand the project society (Alexander & Robertson 2004, Robertson & Robertson 2005). In the onion model, each stakeholder is categorised into zones. The model represents the stakeholders and their involvement with product development. In their scope, stakeholders are mainly requesters and/or decision makers with regard to the requirements specifications. Such stakeholders are selected and analysed by analysts who are positioned inside the system boundary. Therefore, their scope was still inside the system boundary. Pohl (Pohl 2010) proposed the term "context" and introduced a system context that contains all aspects that need to be considered during system development. The world of parties that we introduced in this paper fits well to the Pohl's system context.

Goal-orientated analysis methods are developed to find the alternative requirements in the goal graph (Yu 1997, Chung et al. 1999, van Lamsweerde 2001). When a goal has subgoals with OR-relation, these subgoals are alternative goals to achieve the super goal. If one of the alternative goals is selected, it may be alternated with another goal. We applied the conceptual model to derive the variability of concepts that imply alternative requirements. According to the goal-oriented analysis methods, goals are defined as properties of stakeholders. The scope of these methods is closed inside the world of stakeholders. Our extended the SD model opens the scope of analysts to the world of parties whose changes have an impact on the intention of stakeholders. The dependencies between parties are not based on the personal intention in the original i* framework, but on the structure of organisations.

A power/interest grid, power/influence grid, influence/impact grid, and salience model are techniques introduced in the project management body of knowledge (PMBOK) (Project Management Institute 2013). Stakeholders are grouped according to their power, influence, interest, impact urgency, or legitimacy with regard to the requirements and, each group is placed in a two dimensional space. These techniques are useful in managing stakeholders, but the viewpoint of PMBOK is not concerned with requirements, but the management of stakeholders. We introduced two models, one is an extended SD model, and the other is a static conceptual model, both of which are aimed at the prediction of the requirements changes.

6 Conclusion

We proposed a method that consists of two tools: one is an extended SD model and the other is a conceptual model with a class diagram in UML. The basic concept of the method is that requirements analysts have to take into account the influence on software requirements by social relations in order to identify the volatility of requirements. The scope of the method is not only the world of intentional actors, but also the world of environmental factors. Both worlds are related utilising four relations: sharing, exchanging, ranking, and contracting relations. These relations are interpretations of Fiske's psychological theory. The method was evaluated by applying it into a train service system. As a result, we could report that the method can predict requirements changes within the system. In our future work, we will apply the method to practical examples and refine the method.

Acknowledge

This work was supported by JSPS KAKENHI Grant Number 25330077.

References

Alexander, I. & Robertson, S. (2004), 'Understanding project sociology by modelling stakeholders', *IEEE Software* **21**(1), 23–27.

Bano, M., Imtiaz, S., Ikram, N., Niazi, M. & Usman, M. (2012), Causes of requirement change-a systematic literature review, *in* 'Proc. of Evaluation and Assessment in Software Engineering', The institute of Engineering and Technology, pp. 22–31.

Chung, L., Nixon, B. A., Yu, E. & Mylopoulos, J. (1999), *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers.

Ebert, C. & D., M. J. (2005), Requirements uncertainty: influencing factors and concrete improvements, *in* 'Proc. of the 27th International Conference on Software Engineering', IEEE, pp. 553–560.

Fiske, A. P. (1992), 'The four elementary forms of sociality: Framework for a unified theory of social relations', *Psychological Review* **99**(4), 689–723.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995), *Design Patterns*, Addison-Wesley.

IEC (2001), 'IEC 61882 hazard and operability studies (HAZOP studies) - application guide'.

Kotonya, G. & Sommerville, I. (1998), *Requirements Engineering: Processes and Techniques*, John Wiley & Sons.

MacAulay, L. (1996), *Requirements Engineering*, Springer.

Nakatani, T., Hori, S., Ubayashi, N., Katamine, K. & Hashimoto, M. (2008), A case study: Requirements elicitation processes throughout a project, *in* 'The 16th International Requirements Engineering Conference (RE'08)', IEEE, pp. 241–246.

Nakatani, T., Tsumaki, T., Tsuda, M., Inoki, M., Hori, S. & Katamine, K. (2011), Requirements maturation analysis by accessibility and stability, *in* 'Proc. of the Asia-Pacific Software Engineering Conference 2011', IEEE, pp. 357–364.

Nurmuliani, N., Zowghi, D. & Fowell, S. (2004), Analysis of requirements volatility during software development life cycle, *in* 'The 2004 Australian Software Engineering Conference (ASWEC'04)', pp. 28–37.

Pohl, K. (2010), *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer.

Project Management Institute (2013), *A Guide to the Project Management Body of Knowledge (PMBOK Guide) Fifth Edition*, PMI.

Robertson, S. & Robertson, J. (2005), *Requirements-Led Project Management*, Addison-Wesley.

Sommerville, I. & Sawyer, P. (1997), *Requirements Engineering-A good practice guide*, John Wikey & Sons.

van Lamsweerde, A. (2001), Goal-oriented requirements engineering: A guided tour, *in* 'Proc. of the 5th IEEE International Symposium on Requirements Engineering', pp. 249–262.

Williams, B. J., Carver, J. & Vaughn, R. (2006), Change risk assessment: Understanding risks involved in changing software requirements, *in* 'Proc. International Conference on Software Engineering Research and Practice'.

Yu, E. S. K. (1997), Towards modelling and reasoning support for early-phase requirements engineering, *in* 'The 3rd International Symposium on Requirements Engineering (RE'97)', IEEE, pp. 226–235.

Zowghi, D. & Nurmuliani, N. (2002), 'A study of the impact of requirements volatility on software project performance', *Asia-Pacific Software Engineering Conference* pp. 3–11.