Exploring Risk-Awareness in *i** Models

Constantinos Giannoulis, Jelena Zdravkovic,

Department of Computer and Systems Sciences
Stockholm University
Forum 100, SE-164 40 Kista, Sweden

{cgia, jelenaz}@dsv.su.se

Abstract. The i* modeling technique focuses on an early-phase of requirements engineering aiming at understanding how a system would meet organizational goals, how it would fit within the organizational context, why would it be needed and why should it be preferred among other possible alternatives. Analysts are able to understand early the organizational context that bridges system requirements to organizational goals. However, it is not clear how uncertainty, potential threats and opportunities are taken into account both when developing a strategic dependency model and a strategic rationale model, to facilitate a continuous risk management. This paper proposes a set of guidelines for refining i* models based on risk.

Keywords: i*, risk, goal, dependency, vulnerability, requirements.

1 Introduction

During an early-phase of requirements engineering activities, i* models are used to capture the intentional aspects of a system. What do stakeholders intend to do using the system, how would the system add value to the stakeholders, how would the system support the stakeholders achieve their goals. The objective is to build the context for the system by linking it to the operational and business environment, the organizational structure, to fit stakeholders' expectations and intentions to their goals.

According to [2] echoed by [1], the aim of the early-phase is understand the "whys" of the system requirements, whereas the later-phase is focused on "what" to conclude with requirements specification. Capturing the "whys" provides insights on satisfying the stakeholder's interests, their viability and uncertainty involved.

However, as acknowledged by [3], risks considered early along with stakeholders' goals, can prevent from costs arising from their late discovery (e.g. during or post development) and can contribute good criteria for the analyst to choose among different alternatives when defining requirements. According to [4], risk refers to "... exposure to a proposition of which one is uncertain.", where [4] refers to perceived risk. This definition, from one hand stresses the operational nature of risk relating it to stakeholder's intentions and perception of the organizational environment, while on the other hand, it fits any risk management approach ([5], [6], etc.).

2 Objectives of the Research

In order to capture uncertainty, threats and opportunities during the early-phase of requirements engineering activities we aim at:

- Embedding risks into the development of i* models,
- Linking the early-phase requirements engineering activities and goals set to risk management.

3 Scientific Contributions

Capturing stakeholder interests, as well as how they could be addressed relies on the interaction between analysts, stakeholders and decision makers [1] expressed by the SDM[1] and the SRM[2]. We propose a set of guidelines for refining i* models considering risk through NASA's risk matrix [6], which provides a qualitative understanding of risks, without adding complexity.

To illustrate our guidelines, each step is accompanied with an example coming from a Massively Multiplayer Online Game (MMOG) scenario. There are four actors identified, a game provider (GP), an internet service provider (ISP), a shipping agency and a customer. The GP is the principal actor, creating game content, selling and distributing the game on CDs to customers. The GP obtains the services from an ISP for selling and providing the game. The ISP receives payment as compensation for their service. The GP's game software delivery service takes place through a shipping agency. Customers access the game servers in order to play and pay the GP.

The first step focuses on populating a detailed list of dependencies considering both the SDM and the SRM, focusing on the actor of interest. In step 2 dependencies are analyzed for relevant risks to be identified. Step 3 assesses the identified risks based on their impact and likelihood, resulting into a classification through the risk matrix. Step 4 covers possible influences on the SRM and the SDM coming from different mitigation strategies.

Step 1: Identify detailed dependencies.

The SDM is built where strategic relationships are captured through dependencies. Focus is put on the actor of interest whose dependencies are listed. The analyst builds the SRM for the actor of interest enriching each dependency with intentional relationships. The impact of a dependency within an actor becomes visible, as well as the rationale of the dependency itself. The outcome is a detailed list of dependencies.

In our example, from the SDM, we list the dependencies of the GP. The GP depends on customers for the goal "Game Sales" and the task "Pay for Games", on the ISP for the resource "Hosting Service" and on the shipping agency for the resource "Shipping Service" (figure 1).

[1] The strategic dependency model (SDM) captures the dependency relationships among actors [7].

[2] The strategic rationale model (SRM) captures the rationale behind dependencies and reveals actors' intentions [7].

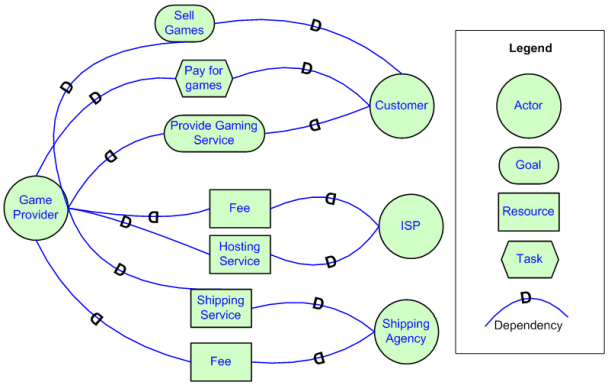

Figure 1: The SDM for the MMOG scenario

The list of dependencies of the GP is enriched with intentional relationships in the SRM (figure 2). The GP depends on:

- Customers for achieving the goal "Game Sales"; this goal can be met by coordinating game provisioning (means-end link), which in our example consists of (decomposition link) the resource Game and the tasks Deliver CDs and Sell Online Gaming,
- Customers for carrying out the task "Pay for games", as the coordinate game provisioning task requires this task be performed,
- The ISP provider for the resource "Hosting service", as the sell online gaming task makes use of it,
- The shipping agency for the resource "Shipping Service", as the Distribute CDs task makes use of it.

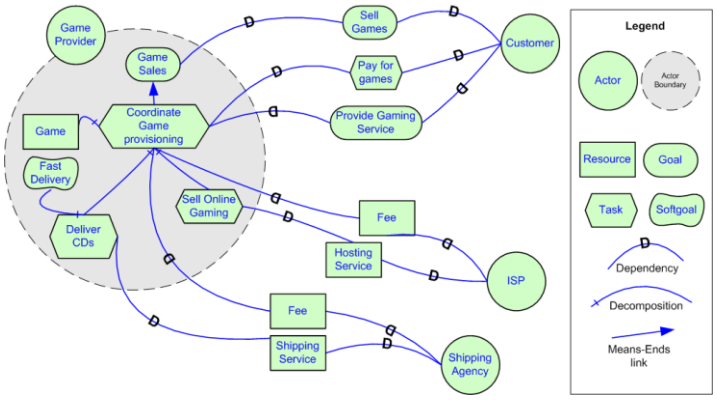

Figure 2: The SRM for the MMOG scenario

Step 2: Identify risks

For each elaborated dependency the analyst aims at discovering, interactively with stakeholders, what could go wrong and what would be the consequences if things

went wrong[3]. Each dependency should be examined to identify whether undesired events could happen, how and when, thus capturing exposure and uncertainty, ergo list risks.

In our example, the GP is running the risk of customers not buying the game (A), or not paying for the game (B), the risk of the ISP provider failing to provide adequate hosting service (C) and the risk of the Shipping agency providing bad service (D).

Step 3: Assess risks

The analyst should assess the impact and likelihood of occurrence for the risks identified. The dependency classification of i* according to vulnerability (open, committed and critical [7]) reflects on impact, whereas for likelihood, probability scales are adequate. The two scales become the two axes of the risk matrix [6] and provide a risk classification.

For our example, considering impact due to vulnerability (1-3), bad shipping service (D) belongs to open (marked with 1), inadequate hosting service (C) belongs to committed (marked with 2) and both not buying (A) and not paying (B) belong to critical (marked with 3). Regarding likelihood, we use a five score scale of occurrence with 0%-20%, 20%-40%, etc. of vulnerabilities being compromised. According to table 1, highest risk lies on (A) and (C)[4].

Table 1: The risk matrix for MMOG

Likelihood				
	5			
	4	**D**	**C**	
	3			**A**
	2			
	1			**B**
		1	2	3
		Impact		

Step 4: Mitigate Risks

The analyst revisits the SRM and considering the matrix starts addressing risks. Mitigating risks should involve stakeholders to help address what if questions relevant to the vulnerabilities. This could include modifying existing intentional relationships within actors, introducing additional to minimize likelihood or make vulnerabilities explicit for the requirement specification or even introducing new dependencies. Changes in the SRM may not necessarily reflect on the SDM (e.g. introducing soft goal decomposition). However, on the later-phase of requirement engineering activities when producing the requirement specification, such changes will appear as additional constraints.

For our example, regarding Hosting service (C), a new soft goal dependency could be introduced between the two actors for Good Online Service. This means the GP

[3] It is not within the scope of this paper to elaborate on the identification of vulnerabilities like [8] and analysis of risks, like [6].

[4] Probabilities identified rely on empirical information coming from stakeholders.

would define what is satisfactory to benefit from the ISP's capabilities. Introducing such a new dependency would result into the modification of the SDM.

An alternative mitigation strategy could be to introduce a soft goal like "Good ISP" for the Sell online gaming task through decomposition. That would serve as a quality goal for the task, and would restrict the selection among alternatives, but would not appear on the SDM as no new dependency is introduced. Ergo, appear as a constraint in the requirement specification for selecting an ISP minimizing exposure, uncertainty or both.

4 Conclusions and Future Work

The proposed guidelines lead to refined i* models, embedding risks qualitatively (exposure and uncertainty). Applied iteratively, the guidelines enhance the use of i* by allowing stakeholders to assess risks related to their goals, elaborate on available possibilities for using information systems and provide risk assessment on using the system to achieve these goals.

5 Ongoing and future work

This paper has presented our effort to embed risks into i* models and provide a link between an early-phase of requirements engineering and risk management. Future work includes examining multi-actor risks and relating i* constructs to risk classifications (e.g. associate critical dependencies to functional requirements, soft-goals to non-functional requirements, etc.).

6 References

1. Yu E.: Towards Modeling and reasoning Support for Early-Phase Requirements Engineering. In: 3rd IEEE International Symposium on Requirements Engineering (RE'97), pp. 226--235. IEEE Press, Washington D.C. (1997)
2. Yu E., Mylopoulos J.: Understanding Why in Requirements Engineering-with an Example. In: Workshop on System Requirements: Analysis, Management, and Exploitation, Schloss Dagstuhl, Saarland, Germany (1994)
3. Asnar Y., Giorgini P., Mylopoulos J.: Risk Modeling and Reasoning in Goal Models. Technical report, DIT-06-008 (2006)
4. Holton G.A.: Defining Risk. J. Financial Analysts, Vol. 60, 6, 19--25 (2004)
5. van Lamsweerde A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, West Sussex (2009)
6. NASA IV&V Management System, Guidelines for Risk Management, S3001, http://www.nasa.gov/centers/ivv/pdf/209213main_S3001.pdf
7. i* Quick Guide, http://istar.rwth-aachen.de/tiki-index.php?page=iStarQuickGuide
8. van Lamsweerde A., Letier E.: Handling Obstacles in Goal-Oriented Requirements Engineering. J. IEEE Transactions on Software Engineering (TSE), Vol 26, 10, 978—1005 (2000)