Itemized Strategic Dependency: a Variant of the *i** SD Model to Facilitate Knowledge Elicitation

Hesam Chiniforooshan Esfahani[1], Eric Yu[2], Maria Carmela Annosi[3]

[1]Department of Computer Science, University of Toronto
[2]Faculty of Information, University of Toronto
[3]Ericsson Software Research, Ericsson Telecomunicazioni S.p.a, Italy
[1]hesam@cs.toronto.edu, [2]yu@ischool.utoronto.ca, [3]mariacarmela.annosi@ericsson.com

Abstract. This paper introduces a variant of the *i** Strategic Dependency (SD) model, called Itemized Strategic Dependency (ISD). The goal of introducing ISDs was to use a simplified version of SD diagrams to model actor dependencies in requirements and process engineering. We used ISD models during the early stages of a software process improvement initiative in one of R&D organizations at Ericsson Italy. In this paper, we explain how ISDs helped us to interact with people who were not familiar with the *i** notation; to elicit their knowledge of organization; and to reveal the hidden problems of their software development process.

1 Introduction

It has been commonly accepted that software development, in many of its aspects, is a human-based activity. The reliance of software companies on the collaboration of project stakeholders (including analysts, designers, developers, customers, etc.) often causes a network of interactions, which its complexity rapidly grows as the size of organizations or projects increase. This phenomenon usually coincides with the emergence of inefficiency symptoms in the process of software development, such as miscommunication of people, loss of knowledge, rework, excessive documentation, and ambiguity of software architecture.

The *i** Strategic Dependency (SD) modeling [1] has been introduced as a way of conceptualizing the collaboration complexities that exist as dependency relations among organizational or system actors. SD models have been used in Requirements Engineering (RE) and Process Engineering (PE). While *i** modeling (including SD and SR) can be used by specially-trained analysts, more effective modeling and knowledge elicitation can be achieved when domain stakeholders are able to directly contribute to the construction and analysis of such models. When following the original format of SD models [1], as the number of actors and dependencies increases, the diagram becomes cluttered and cumbersome to extend or modify. In this paper we introduce a variant of the SD model, called *Itemized Strategic Dependency* (ISD), in order to promote the process of knowledge elicitation during the early stages of RE and PE, and to facilitate the understandability of models by those who are not familiar with the *i** notation. ISD models have been successfully used in a Software Process

Improvement initiative that we are currently involved in one of the R&D units at Ericsson Italy.

2 Objectives of the Research

As mentioned before the main objective of this research is to promote the understandability of the *i** SD models, by simplifying the representation style of strategic dependencies. To achieve this objective, we need to answer the following two questions:

1. What needs to be retained in the simplified SD, and what can be omitted?
2. What are the tradeoffs of the new approach (e.g. what info can be lost)?
3. How will the new approach work out in a real life project?

3 Itemized Strategic Dependency Models

The complexity of *i** SD models happens when the number of dependums between actors increases. The ISD models are introduced to simplify the visual representation of SD models, while conveying the same information. In such models, a single dependency link is used to represent all the dependencies from one actor to another in one direction, with the dependums written as an itemized list associated with the link. Figure 1 and 2 provide two examples. The dependency link is a continuous curved line from the depender actor to the dependee actor with a single "D" near the middle. Unlike the original SD, the dependums are not enclosed in different shapes according to dependum types.

To further simplify modeling and to support incremental elicitation, we define two submodels of the ISD: *Functional ISD*, for representing functional dependencies; and *Quality ISD*, for representing the quality attributes of dependency relations. We also define a concept of the *Viewpoint Actor* (VA), the actor from whose viewpoint the model is constructed. The viewpoint actor is denoted by a thicker circle. For an ISD with a viewpoint actor, relationships among other actors are likely to be sketchy and incomplete. This form of the ISD is intended especially for interacting with stakeholders during individual interviews. When no viewpoint actor is indicated in an ISD model, the model represents the understanding of the analyst/modeler, typically gained by integrating the viewpoints of all actors.

3.1 Functional Itemized Strategic Dependency (FISD)

Each FISD shows all the functional dependencies that a Viewpoint Actor has with other organization actors. Here, by *functional dependency* we mean all dependency relations that are not related to any particular quality attribute. Such dependency relations would be Resource, Task, or Goal dependencies in an original SD representation format. As shown in Figure 1 each dependency relation contains a list of dependency items, for which the SA depends on other organizational actors, or vice

versa. For instance, in our case study in Ericsson, Designers were depending on System Manager for the Node Requirements Specification (NRS), Feature Specification, and Pre-study documents; also for setting up meetings to clarify these documents requirements.

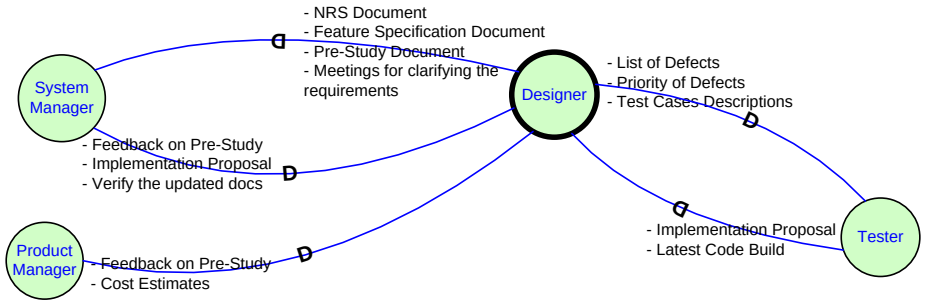

Figure 1: Sample FISD, representing the functional dependencies of Designer (the Viewpoint Actor) and System Manager

If we wanted to represent these dependency relations in original SD models we had to depict 14 dependency links. For instance, just for representing the SD relations of Designers to System Manager we had to draw three resource dependencies for three documents, and one task dependency relations for setting up the meeting. In should be mentioned that since the sample FISD in Figure 1 is developed from the viewpoint of designers, the represented dependencies of the System Manager (or other actors) to Designer is the perception of designers, not necessarily in agreement with System Manager's perception. The complete FISDs of an organization can be developed by aggregating the VP-based FISDs, developed for each organizational actor.

3.2 Quality Itemized Strategic Dependency (QISD)

QISDs represent more delicate aspects of dependency relations, by listing the quality attributes of the functional dependencies represented in FISDs. In other words, QISDs represent the Softgoal dependency relations, which are related to the functional dependencies identified in FISDs. Every FISD can be transformed to a number of QISDs, each representing the pair-wise dependency relations of the VA and a subset of other actors. Figure 2 shows a QISD we developed for our case study, and reveals some of the quality attributes that are expected from dependums of dependency relations between Designer and System Manager. For instance, it shows that designers expect that the NRS documents be sufficiently detailed, technical, and regularly updated. On the other hands, System Manager expects that the designers' feedback on Pre-Study document to be reliable and accurate.

As shown in the Figure 2 for every functional dependency (represented in FISD models) there is a corresponding entry in the QISD models. If the quality attributes of that functional dependency were already extracted, they were written on the dependum list, otherwise, a number of question marks represent the fact that the quality attributes of the corresponding functional dependency have not been yet

identified. Since the process of knowledge extraction both in requirement and process engineering is usually iterative, this approach can guide modelers in better clarification of the complexities of a subject domain.

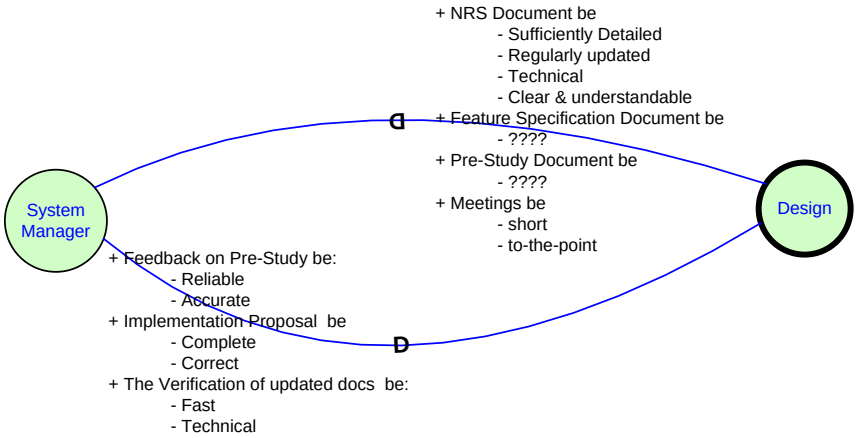

Figure 2: Sample QISD, representing the quality attributes that are important on dependency relations of SA:Designer and System Manager

4 Conclusions

This visual design of ISDs greatly reduces the number of lines on the dependency diagram. The itemized text block of dependums can be easily edited or added to. The main drawback is that the block of dependums is now visualized as a single unit, while semantically each item should be treated as independent. It is therefore harder to visualize redirecting one of the dependums to a different actor, e.g., in cases where a mistake was made, or when considering alternative configurations during process redesign. Further, it is no longer possible to interleave dependency links going in opposite directions to group related links together. One possible solution to this limitation is the use of tables, instead of text blocks on dependency links. In this way further information about dependums (e.g. their types) can be expressed.

We used the ISD models for the early phase on a Software Process Improvement (SPI) initiative in one of the R&D units of Ericsson in Italy. The primary objective of that phase of the SPI initiative was to identify the problematic issues of the current process, in order to come up with proper solutions in the later phases of the initiative. To achieve this goal, we conducted two rounds of interviews with operative and managerial personnel of that R&D unit. In the first round of interviews, we asked the interviewees to describe their role and responsibilities in the organization, and explain different kinds of collaborations that they had with other organization role. After the first round of interviews we gained an initial understanding of the unit, thus we started to develop FISD and QISD models. During the second round of interviews, we used

the initial models and asked interviewees to complete the initial models. Using the ISD models for structuring the interviews we could guide the interviewees to:

1. Visually observe their collaborations with other organizational actors, in terms of the mutual dependency relations.
2. Validate our initial understanding of dependency relations in that R&D unit.
3. Express the functional or quality dependencies that they did not expressed during the first round of interviewees.
4. Identify process problems, which were due to the quality attributes associated with functional dependencies, and were not expressed at the first meetings.
5. Identify process problems, which were due to missing dependencies (i.e. dependencies that should have been exit for facilitating the work).
6. Identify process problems, which were due to unnecessary collaborations and dependencies.

5 Ongoing and future work

As the ongoing project we are still involved in the SPI initiative. We have almost completed the preliminary stage of this initiative, and with the help of ISD models gained a good understanding of the collaboration complexities of the R&D unit. We are going to integrate the information we collected from different Viewpoint Actors, and build a set of comprehensive dependency models that represent the as-is dependency structure of the R&D unit. These models will be used to explicitly represent the hidden or unnecessary complexities, which reduced the productivity of that unit.

As of a future work, we are going to integrate this modeling approach as part of a method engineering framework introduced in [2]. Besides, we are working on a comprehensive SPI framework, which is based on the intentional aspects of development processes, and works with regard to the functional and quality goals of software processes. We planned to use ISD models to represent new dependency relations that will be proposed as the to-be process in an SPI initiative. The models in this paper were developed using Microsoft Visio. We hope to extend the OpenOME in near future to support ISDs as well.

Acknowledgment. We would like to greatly appreciate the constructive comments of reviewers of this paper.

References

1. E. Yu. *Towards modelling and reasoning support for early-phase requirements engineering*. In Proceedings of the Third IEEE International Symposium on Requirements Engineering: (received "Most Influential Paper After 10 Years Award" at RE'07) (1997)
2. H. Chiniforooshan Esfahani and E. Yu. *Situational Evaluation of Method Fragments: an Evidence-Based Goal-Oriented Approach*. In 22nd International Conference of Advances Information Systems Engineering (CAiSE'10). Tunisia (2010)