Towards Aspectual i*

Fernanda Alencar[1] , Jaelson Castro[1], Cleviton Monteiro[1],
Ricardo Ramos[1], and Emanuel Santos[1]

[1] Universidade Federal de Pernambuco, Recife, Brasil
{fernanandaalenc, cleviton}@gmail.com, {jbc, rar2, ebs}@cin.ufpe.br

Abstract. The i* framework is a very popular approach in the Requirements Engineering community. However, crosscutting concerns are not handled explicitly in i*, compromising the modularity and consequently the complexity and evolution of such models. To deal with the complexity of i* models, the use of structuring mechanisms has been investigated. Moreover, the i* framework has been extended to incorporate the principles of aspect orientation, aiming at handling explicitly crosscutting concerns in i* models. Some metrics have been created to evaluate the understandability of requirements modeled using both the i* framework and the aspectual i*.

Keywords: requirements, modularity, graphical complexity, evolution, understandability, metrics, aspects.

1 Introduction

During the early stages of requirements engineering, it is necessary to identify and specify the stakeholders' needs. The i* framework provides expressive models to achieve this, wherein motivations and rationale are explicitly captured in a requirements model. Thus, the i* framework is becoming widely used by the Requirements Engineering community, capturing social and intentional characteristics of the system organisation context.

However, i* models, even for small applications, may become cluttered, compromising their evolution, scalability, and, consequently, its understandability. In large and complex projects, this problem increases significantly. Furthermore, traditional i* models tend to include scattered and tangled representations that are hard to understand and maintain. Therefore, as systems models evolve and grow in scale and complexity, better encapsulation and localization primitives are needed. Thus, we proposed to use some of the Aspect Oriented Software Development concepts to improve the i* models views.

The paper is organised as follows. Section 2 describes the objectives of the research. Section 3 presents the proposed tool to automate the identification of the candidate aspects on i* models. Finally, Section 4 concludes the work and points to future research.

2 Improving the understandability of the i* models

Since 1997 we have been investigating the use of i*, aiming at identifying its strengths and weaknesses. In the last ten years, this research group has proposed several approaches either to improve i* or to integrate it with other relevant techniques. For example, approaches have been proposed to reduce the gap between organizational and functional requirements by integrating i* and UML models [1,2]. To deal with the complexity of i* models, the use of structuring mechanisms has been investigated. Moreover, the i* framework has been extended to incorporate the principles of aspect orientation, aiming at handling explicitly crosscutting concerns in i* models.

We proposed an approach for identifying and separating crosscutting concerns (henceforth referred to as candidate aspects) in i* models [2, 3]. In particular, we provide means to discover and model tangled and scattered tasks, goals and softgoals as well as internal links (means-ends and contributions links). Composition is handled in a graphical way. Hence, we introduced (i) a set of guidelines to identify the candidate aspects in i* models; (ii) an extension of the i* metamodel by adding aspectual constructors to modularize candidate aspects and its composition with other system modules; and (iii) the concrete syntax of the aspectual i* modelling language (Fig. 1).

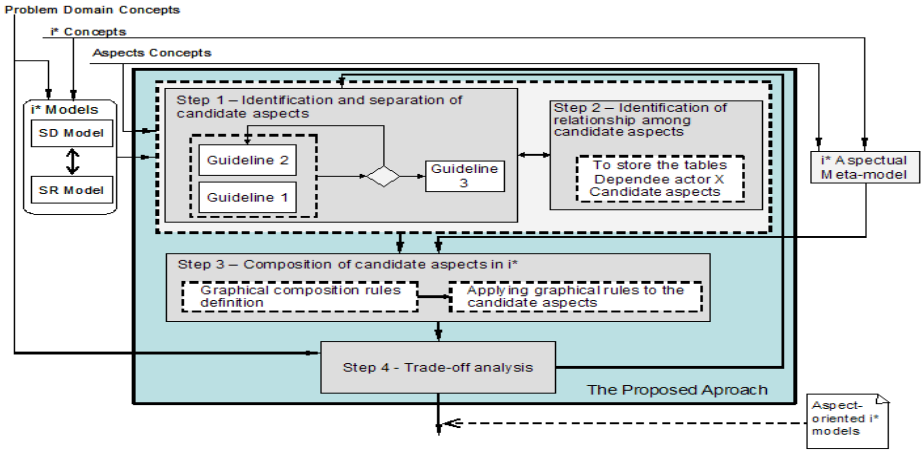

Fig. 1. The overview of the approach

We also evaluate the requirements document quality, finding pieces that can be improved with the application of requirements patterns and refactorings [5,6,7,8]. The refactoring opportunities that we have identified have the goal of making the requirements more understandable as well as to improve the overall organization of the project. But depending on the system quality attributes considered (for example: reusability, maintainability, etc) the requirements engineer might take actions that slightly differ from the guidelines proposed here. Also, the expected granularity of the requirements descriptions might influence the use of the refactoring opportunity. The requirements engineer should take these issues into consideration.

3 The iAspectPlugin

It is quite hard to manually identify crosscutting concerns directly from the organizational i* models. Thus we propose some means to automate this process. Therefore, the plug-in was developed with the objective of turning the identification faster and easier. The current version of iAspectPlugin [4] does not automates the whole process described in [3] yet; it only supports the activities of identifying and representing the crosscutting concerns.

When we developed the iAspectPlugin we chose to reuse the OME environment instead of creating a new tool exclusively to treat the crosscutting concerns identification, because this is a stable tool, largely used and allows a simple way to be extended.

Regarding the iAspectPlugin graphical interface, Fig. 2 shows a screen captured from the OME menu with the developed plug-in already installed. The circle is highlighting the button that was added to run the crosscutting concerns identification automatically. At all five buttons were added by the plug-in and are numbered in Fig. 2. The buttons 1, 2, 3 and 4 can be used to construct manually i* models with aspects. The user can do the organizational modeling with any i* element (Actors, intentional elements, links, etc) and at any time press the button 5 to run the crosscutting concerns identification. Then the plug-in performs the aspect identification and updates the model creating the crosscutting concerns stereotypes, composition rules inside the aspects and crosscuts links. It also moves the elements that were captured by the guidelines to the crosscutting concern internal representation (expansion).

A plug-in for the OME tool is a Java class that implements the OMEPlugin and interacts with other classes defined by the plug-in developers. The Java classes design's definition started using the GRASP (General Responsibility Assignment Software Patterns) guide for the requirement analysis.

Fig. 2. OME menu with the buttons added by the plug-in

4 Conclusions and future works

In order to improve the evolution, reuse and maintainability of i* models we have considered the modularization and localization of concerns. In particular, guidelines are required for the identification of crosscutting concerns. Moreover, we also need a notation for the description of improved models as well as compositions rules to keep a record of the relationships between the crosscutting concerns and the other model elements. The approach aims at reducing the graphical complexity of large i* models.

We also extended the i* metamodel to introduce two main new concepts: aspectual actor and crosscut relationship. Aspectual actors modularize candidate aspects and the crosscut relationship captures the information of source and target model elements, as well as, when and how an aspect crosscuts other model elements.

Currently, we are working on the inclusion of routines identification and their association to concerns which may be tangled and scattered across various actors in i* models. Another point to be investigated is how to deal volatile concerns that might not be necessarily scattered or tangled. Also some metrics have been proposed to evaluate the understandability of requirements modeled using both the i* framework and the aspectual i. Finally, we plan to redefine the current tool [4] to support the new guidelines and the entire process for identification and separation of candidate aspects.

References

1. Castro, J., Alencar, F., Cysneiro Filho, G.., Mylopoulos, J. Integrating Organizational Requirements and Object Oriented Modeling In: 5th IEEE International Symposium on Requirements Engineering, 2001, Toronto. Proceeding of 5th IEEE International Symposium on Requirements Engineering. IEEE Press, 2001. p.146 – 153.
2. Alencar, F., Cysneiros Filho, G.., Castro, J..: Support for Structuring Mechanism in the Integration of Organizational Requirements and Object Oriented Modeling. In. V Workshop de Engenharia de Requisitos (WER02). Espanha. Valência: Anais/Workshop de Engenharia de Requisitos (2002).
3. Alencar, F., Castro, J., Moreira, A., Araújo, J., Monteiro, C., Ramos, R., Mylopoulos, J.: Simplifying i* Models. Agent-Oriented Information Systems Workshop Proceedings (AOIS'07), Intl. Conf. on CAiSE'07. (eds.) Tapir Academic Press, pp. 635-649, Norway (2007).
4. Monteiro, C., Ramos, R. A., Castro, J. B., Araújo, J. Moreira, A., Alencar, F.: The iAspectPlugin to Automatize the Identification of Crosscutting Concerns on i* Models. In: I Latin-American Workshop on Aspect-Oriented Software Development 2007, João Pessoa. I Latin-American Workshop on Aspect-Oriented Software Development. João Pessoa : pp. 105-116, Sociedade Brasileira de Computação, (2007).
5. Ramos, R. A. ; Alencar, F. ; Araújo, J. ; Moreira, A. ; Castro, J. and Penteado, R.: i* with Aspects: Evaluating Understandability. In: Workshop on Requirements Engineering, 2007, Toronto, Canada (2007b).
6. Ramos, R.A., Araújo, J., Castro, J., Moreira, A., Alencar, F., Silva, C.: "Uma abordagem de instanciação de métricas para medir documentos de requisitos orientados a aspectos", in: 3º Brazilian Workshop on Aspect Oriented Software Development - WASP2006. Florianopolis, Brasil (2006)
7. Ramos, R. A.; Araújo, J. ; Moreira, A. ; Castro, J. ; Alencar, F. and Penteado, R.: "Um Padrão para Requisitos Duplicados", in: 6th Latin American Conference on Pattern Languages of Programming (SugarLoafPlop'2007), Porto de Galinhas, Recife, Pernambuco, Brasil (2007a)
8. Ramos, R., Piveta, E., Castro, J., Araújo, J., Moreira, A., Guerreiro, P., Pimenta, M., and Tom Price, R.. Improving the Quality of Requirements with Refactoring. In: VI Simpósio Brasileiro de Qualidade de Software – SBQS2007, Porto de Galinhas, Recife, Pernambuco, Brasil, Junho 27 – 30 (2007c)