The JGOOSE tool

Gustavo Cesar Lopes Geraldino and Victor Francisco Araya Santander

Universidade Estadual do Oeste do Parana, UNIOESTE Cascavel, PR, Brasil
gclgeraldino@gmail.com and victor.santander@unioeste.br

Abstract. Requirements engineering is an important phase of software engineering. The integration of the various requirements models remains a major challenge. In previous work we proposed a process to generate UML Use Cases from i* (iStar) organizational models. It is supported by the JGOOSE tool, which transforms i* models into Use Case diagrams and textual descriptions. This paper presents this tool including the E4J i* (Editor for i*) and E4J Use Cases (Editor for Use Cases).

Keywords: i* Framework, Requirements Engineering, Use Cases.

1 Introduction

System development should occur in a context where organizational processes are well established. The primary reason for software system failure is the lack of proper understanding of the organization by the software developers [1]. Unfortunately, the dominant object-oriented modeling technique, UML, is badly equipped for organizational requirement modeling. We need other techniques, such as i* (iStar) [1] to represent organizational modes. The i* (iStar) framework is well suited to represent the organizational requirements that occur during the early requirements phase. It is capable of modeling variability and offers a rich set of modeling concepts, such as goal, softgoal, task and resource. These early activities can enable an understanding of how and why the requirements emerged.

Nevertheless, organizational requirements must be related to functional requirements which are often represented as Use Cases [2]. However, Use Case development demands great experience of the requirements engineers. The heuristics presented in the literature to develop Use Cases are not sufficient to allow a systematic development. In this sense, in previous work [3] we proposed an approach to support the development of Use Cases from organizational models described in i*.

Our approach allows the elicitation and specification of requirements from the actors goal in relation to the system-to-be. Moreover, the relationships between system to be and its environment can also be expressed in terms of goal-based relationships obtained from the organizational models. Hence, we can derive and map goals, intentions and motivations of organizational actors to main goals of Use Cases.

In order to support the mapping of the i* into Use Cases models we developed the JGOOSE tool (Java Goal Into Object Oriented Standard Extension) [4]. Using this tool we can derive Use Cases from the TELOS (.tel) [5] [6] files generated by OME (Organization Modelling Environment) [7] tool and from i* models developed using the E4J i* (Editor for i*) [8]. The E4J i* is integrated into JGOOSE. More recently,

we have developed the E4J Use Cases (Editor for Use Cases) [9]. This editor is also integrated to the JGOOSE.

This integration turned JGOOSE into a standalone application, dispensing the need to have an external software (extra tool) to create i* models and to edit use cases generated from i* models. Hence, it also eliminates the dependence on the external tools such as OME. Therefore, in this paper we present the JGOOSE tool including the E4J i* and E4J Use Cases editors. It is important to notice that the current version of JGOOSE also support deriving use cases from BPMN (Business Process Model and Notation) [13] models [10]. However, this functionality will not be present in this paper.

The remaining of this paper is organized as follows. Section 2 introduces the concepts used by i* framework to represent organizational requirements and early requirements. In Section 3 we describe the JGOOSE tool and the E4J i* and E4J use cases editors. In Section 4 we present a brief example using the JGOOSE. Section 5 concludes the paper.

2 i* Framework

When developing systems, we usually need to have a broad understanding of the organizational environment and goals. The i* framework [1] provides understanding of the reasons (Why) that underlie system requirements. It focuses on strategic actor relationships. i* allows the description of the intentions and motivations involving actors in an organizational environment. It offers two models to represent these aspects: The Strategic Dependency (SD) Model and the Strategic Rationale (SR) Model.

The SD focuses on the intentional relationships among organizational actors. It consists of a set of nodes and links connecting them, where nodes represent actors and each link represents the dependency between actors. The depending actor is called Depender and the actor who is depended upon is called Dependee. The i* framework defines four types of dependencies among actors: goal, resource, task and softgoal. The SR model complements the SD model. It supports the modeling of the reasons associated with each actor and their dependencies. SR model assists in requirements engineering by allowing process elements and the rationales behind them to be expressed.

During early requirements engineering, the SR model can be used to understand how systems are embedded in organizational actors routines, to generate alternatives, as well as to support the reasoning that goes along the choice of alternatives.

3 JGOOSE

The JGOOSE tool [4] supports the guidelines proposed in [3] for deriving Use Cases from organizational modeling. The results of the integration processes are Use Case diagrams for the intended system and textual descriptions scenario for each Use Case using the Cockburn template [11]. Use Cases are derived considering the intentions associated to the actors of the organizational environment.

This tool has undergone several evolutions over the last years. The first version was presented in [14] in which it was allowed to derive use cases from i* models

developed in the OME tool [7]. In this first version there was only an automated process of transformation not allowing to edit the i* models and the use cases. Then, as an evolution of the tool was incorporated an editor for i* models named E4J i* [8] which allows to create and modify SD and SR models and use them to generate the textual and diagrammatic use cases in the same environment of JGOOSE.

Later, a use case editor named E4J Use Cases [9] was added to this environment, which allows to create and modify use case diagrams. More recently, it was added the editor E4J BPMN and the module BP2UC (Business Process to Use Cases) [10]. The editor allows to construct and modify BPMN models and the BP2UC module automates the guidelines proposed in [10] generating use cases. The JGOOSE main screen is shown in figure 1b.

3.1 E4J i*

The E4J i* is a graphical editor for the i* models. It supports the creation and modification of SD and SR models as well as the mapping of parts of these models to the JGOOSE tool internal components. This new extension to JGOOSE allows the derivation of the Use Cases without the need of any external tool.

According to [1], the SD and SR models are composed by graphs with vertices and edges. Therefore, we can reuse a module or library that supports graph edition. In this sense, we relied on the JGraphX [8] Java library to represent the i* framework models. This library also has been used for the E4J Use Cases development. More details about the E4J i* can be seen in [8].

3.2 E4J Use Cases

E4J Use Cases is a use cases diagram editor, which provides features and functionality for creating and manipulating use cases diagrams generated by using the JGOOSE. Since E4J Use Cases is integrated with JGOOSE, the conversion of structures and templates from JGOOSE to E4J Use Cases is done via internal routine without the need for intermediate files. E4J Use Cases generates use case diagrams based on the use cases mapped by JGOOSE.

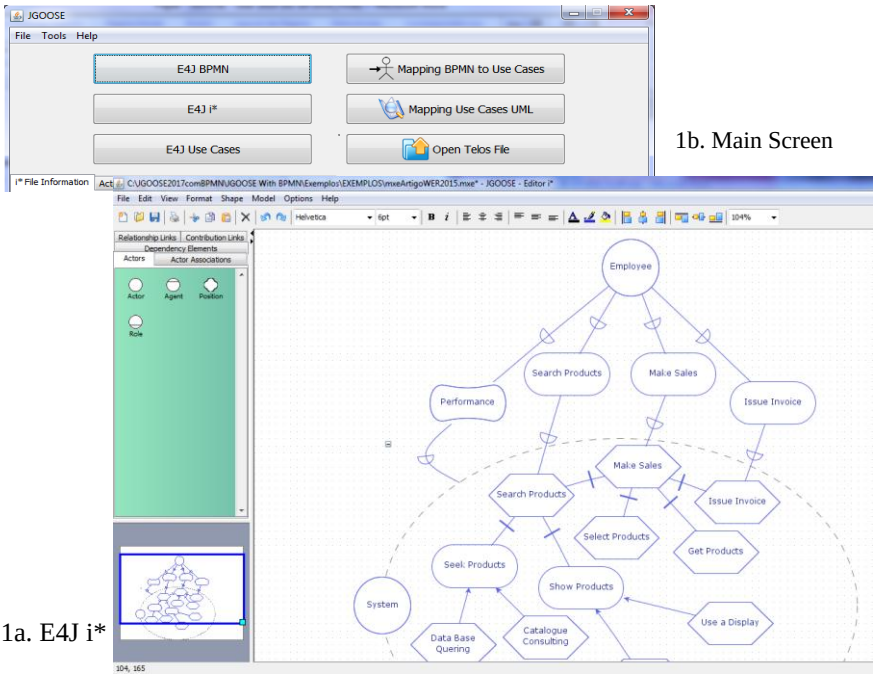
Basically, the editor generates one element (ellipse containing the name of the use case) for each use case and one element (“doll” containing the name of the actor) for each actor mapped by JGOOSE and one link (association type) between each actor and their use cases. In addition, E4J Use Cases verifies if each step in the textual descriptions of each use case is a use case in itself, and if so, that use case is present in the diagram and a link (of type «include ») is generated between the use case being verified and the use case that represents the step.

Along with the verification of the steps, it is checked if each step has extensions and if these are use cases by themselves. If so, a («extend») link is generated between the use case being checked and the use case representing the extension.

More details of the E4J Use Cases can be consulted in [9].

4 Using JGOOSE

This Section relies on an example to describe the use of the tool. Note that the i* organizational models, developed in E4J i*, are mapped to the classes and objects of the JGOOSE tool. After this mapping, the user may generate the Use Cases from models developed in E4J i*. E4J i* interacts with the user by means of its graphical view, which includes many resources for model manipulation and edition (see figure 1a). The example in the sequel displays the screens, windows, and dialogues of the tool. We use the E4J i* to create the model of a generic organization, which needs an application to support its buying and selling process. This example was borrowed from [4].



1b. Main Screen

1a. E4J i*

Fig. 1. Screens of JGOOSE (1b. Main Screen and 1a. E4J i*)

The building of the SD and SR models can be summarized as addition of elements to the palette and linking among these elements. After building the SD and SR models in E4J i*, we can save them in the .mxe format (native of JGraphX java library), export them to the iStarML [15] format, as well as use them to generate the Use Cases in UML. In order to generate the Use Cases from an E4J i* model, follow the steps: (1) click the File menu and (2) select the Generate Use Cases submenu (see figure 1a). After this, a JGOOSE interface will be activated (see figure 1b).

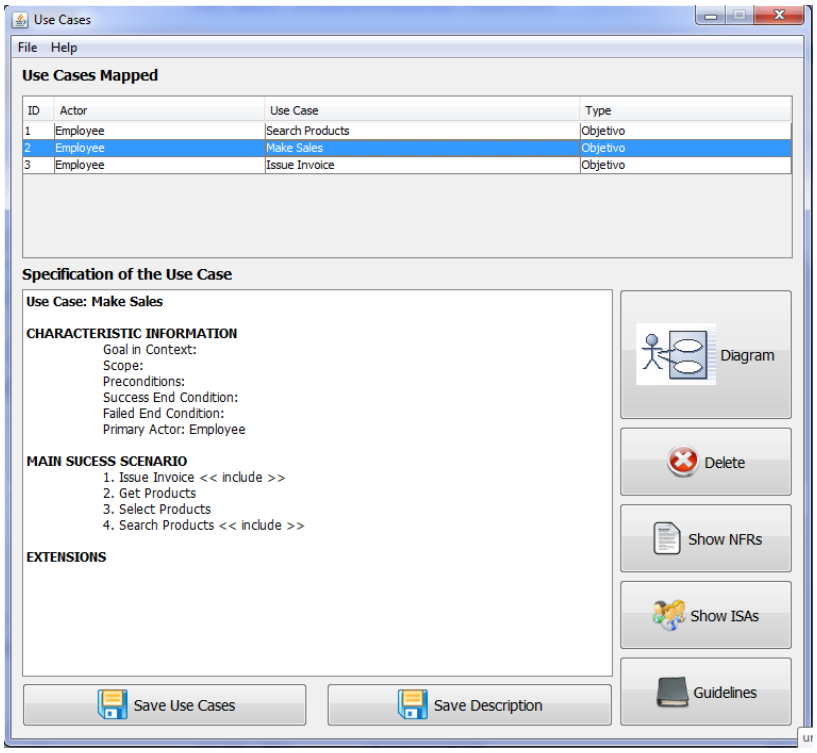


Fig. 2. Screen of JGOOSE - Intermediary Screen for generating use cases from E4J i*

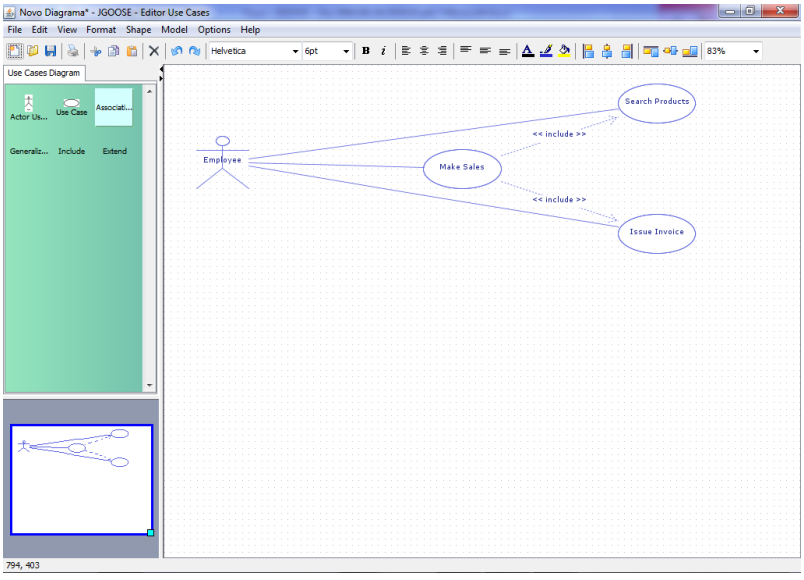


Fig. 3. Screen of JGOOSE - Main Screen of E4J Use Cases

When JGOOSE screen is enabled, the tool asks the user what is the actor that represents the system. After this selection, it is possible to visualize information about the opened file and existing actors, mapped elements (goals, resource, softgoal, task) and links (dependence, means-end, task-decomposition and ISA relation). The next step is to generate the UML Use Cases. When the user clicks the Mapping Use Cases UML button (see figure 1b), it activates the screen showed in figure 2. In this example, it is possible to visualize that the unique actor mapped to the diagram is Employee and the Search Products, Make Sales and Issue Invoice Use Cases are associated to the actor, meaning that there are dependencies between these actors and the system. Note that the textual description for each use case using the Cockburn [11] template can be edited and saved. Pressing the Diagram button, the use case diagram is open in the E4J Use Cases editor (see figure 3).

5 Conclusion

In this article, we presented the JGOOSE tool. This tool is integrated by three editors E4J i*, E4J Use Cases and more recently by the E4J BPMN not presented in this paper. The integrated editors turned JGOOSE into a standalone application, dispensing the need to have an external software (extra tool) to create i*, use cases or BPMN models or diagrams. However, it is important to notice that the main motivation for the JGOOSE development was the need to support the guidelines proposed in [3] for deriving use cases from i* models. So, the integrated editors support this process.

In the last years the JGOOSE have been used to teach undergraduate students into requirements engineering courses. Our experiences using the JGOOSE on the requirements engineering education are related in [16] [17]. Nowadays, we are preparing controlled experiments (using Experimental Software Engineering principles) with students to evaluate the JGOOSE for deriving use cases from i* and BPMN models. Some hypothesis has been established and will be refuted or supported considering the experiment results.

As future works, we plan to improve the import routine that deals with iStarML [15] structure. Thus, soon E4J i* will be able to import i* models in this format. Furthermore, we also want to create i* organizational models, which comply with the syntax and semantic constraints proposed in iStar 2.0 [18]. Similarly, we intend to provide the changes in textual and graphical representation of Use Cases generated by JGOOSE to be reflected in the SD and SR models associated and built in E4J i*. For this, we will consider the guidelines proposed in [19]. Another future work involves the use of JGOOSE in industrial case studies. Finally, we emphasize that the tool is open-source code and is available for download (with a user manual in Portuguese) in [12]. Also a video (in Portuguese) is available in [20].

References

1. Yu, E. S-K. Modelling Strategic Relationships for Process Reengineering. Phd Thesis - University of Toronto, 1995.
2. Booch, G., Rumbaugh, J., Jacobson, I. UML: User Guide, 2 Edition, Campus, 2005.

3. Santander, V. F. A., Castro, J. F. B. Deriving Use Cases from Organizational Modeling. In: IEEE Joint International Requirements Engineering Conference - RE, 2002.
4. Brischke, M., Santander, Victor F. A., Silva, I. F. Melhorando a Ferramenta JGOOSE. In: 15th Workshop on Requirements Engineering, 2012, Buenos Aires, 24 a 27 de Abril. Anais do 15th Workshop on Requirements Engineering, 2012. In Portuguese.
5. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M. Telos: Representing knowl-edge about information systems. ACM Transactions on Information Systems (TOIS), ACM, v. 8, n. 4, p. 325362, 1990.
6. Koubarakis, M., Mylopoulos, J., Stanley, M., Borgida, A. Telos: Features and formalization. [S.l.]: Computer Science Institute, Foundation of Research and Technology, Hellas. 1989.
7. Yu, E., Yu, Y. Organization Modelling Environment. Available in: http://www.cs.toronto.edu/km/ome/. Last Access: July, 31 2019.
8. Merlin, L. P. et al. Integrating the e4j editor to the jgoose tool. In: Anais do WER15 - Workshop em Engenharia de Requisitos, Lima, Perú, April 22, 23, and 24, 2015.
9. Peliser, D.; Santander, V. F. A.; Freitas, I.; Andrade, S. C.; Schemberger, E. E4J Use Cases: um editor de diagrama de casos de uso integrado à ferramenta JGOOSE. In: 35th International Conference of the Chilean Computer Science Society (SCCC 2016), Valparaíso, Chile. NY 12571 USA: IEEE Catalog Number CFP16139-ART, 2016
10. Girotto, A. N.; Santander, Victor F. A ; Silva, I. F. ;Toranzo, Marco. A.; Uma proposta para derivar Casos de Uso a partir de modelos BPMN com suporte computacional. In: 36th International Conference of the Chilean Computer Science Society (SCCC 2017), 2017, Arica, 16 a 20 de Outubro.
11. Cockburn, A. Writing effective Use Cases. [S.l.]: Addison-Wesley Reading. 2001.
12. Software Engineering Laboratory - UNIOESTE. Available in: http://inf.unioeste.br/les/. Last Access in: July, 25 2019.
13. BPMN. Business Process Model and Notation Version 2.0.2. [S.l.], December 2013.
14. Brischke, M., Santander, Victor Francisco Araya., Castro, J. F. B., GOOSE: Uma Ferramenta para Integrar Modelagem Organizacional e Modelagem Funcional In: Jornadas Chilenas de Computación - V Workshop Chileno de Ingeniería de Software, Valdivia, Chile. (2005).
15. Cares, C., Franch, X., Perini, A., Susi, A. iStarml: An xml-based model interchange format for i*. In: Proc. 3rd Int. i* Workshop, Recife, Brazil. [S.l.: s.n.], v. 322, p. 1316. 2008.
16. Santander, Victor F. A. Avaliando a utilização da Técnica i* no Processo de Ensino e Aprendizagem na Engenharia de Requisitos - Um Relato de Experiência. In: IV Fórum de Educação em Engenharia de Software, XXV Simpósio Brasileiro de Engenharia de Software (SBES), São Paulo, 2011.
17. Santander, Victor F. A.; Silva, I. F. Avaliando a utilização da Ferramenta JGOOSE no Processo de Ensino e Aprendizagem na Engenharia de Requisitos: Um Relato de Experiência. In: XIX Conferência Internacional sobre Informática na Educação, 2014, Fortaleza.
18. Dalpiaz, F., Franch, X., Horkoff, J.: iStar 2.0 Language Guide. Available in https://arxiv.org/abs/1605.07767 (2016).
19. Bhuiyan, Moshiur & Haque, Farzana & Shabnam, Luba. (2018). Integration of Organisational Models and UML Use Case. Journal of Computers. 13. 1-17. 10.17706/jcp.13.1.1-17.
20. https://youtu.be/FdDaQp_XqFQ. Last Access in: July, 31 2019.