Specifying guidelines to transform i* Model into User Stories: an overview

Celso Agra[1], Aeda Sousa[1], Josenildo Melo[1], Márcia Lucena[2], Fernanda Alencar[3]

[1]Programa de Pós-Graduação em Engenharia da Computação – UPE
{clasf, amcs, jasm}@ecomp.poli.br
[2]Departamento de Informática e Matemática Aplicada – UFRN
marcia.lucena@gmail.com
[3]Departamento de Eletrônica e Sistemas - UFPE
fernandaalenc@gmail.com

Abstract. Requirements engineering is a process of obtaining and detailing the goals of systems-to-be. Therefore, different approaches can improve the requirements elicitation phase, such as diagrams and documentations. Scenarios based on GORE approach, such as i* framework, model the early phase of requirements stage. The framework, however, can become a large and unreadable diagram. On the other hand, user stories are clear, concise, and can represent the late-phase of requirements. Therefore, this paper presents an overview of a proposal to transform i* models into user stories, as a way to facilitate the management of requirements in the agile projects.

Keywords. i* Framework, Goal-Oriented Requirements Engineering, Scenario transformation, User stories, Agile projects

1   Introduction

Requirements engineering is a process of obtaining and detailing the goals of systems-to-be [1]. A set of different solutions were created to improve the requirements elicitation phase, such as diagrams and documentations artifacts. They are scenario techniques based on Goal-Oriented Requirements Engineering (GORE) modeling which are being widely studied by academics and practitioners. These techniques define goals to stakeholders clearly and easily, and thus, they are becoming useful for Requirements Engineering [4].

According to [2], the main challenges found in Requirements Engineering are related to lack of communication and poor quality of artifacts. Communication problems during the requirements elicitation phase can compromise the schedule or even the budget of a project. The use of scenarios, such as GORE models and other artifacts requirements, may contribute with the elicitation and documentation requirements, and also with the building of better quality software.

Scenarios based on GORE approach, such as i* framework, model the early phase of the requirements stage, and consider organizational contexts, intentions and ration-

ales for stakeholders demands [5]. On the other hand, user stories are simple, clear and concise artifacts used to represent and detail requirements, and are, thus, valuable to users [6]. In addition, user stories describe functionalities in the late requirements phase, during the software lifecycle, considering "how" those requirements should be developed.

A set of proposals were made by combining i* model with other late-phase approaches and subsequent phases of software lifecycle [5]. However, none of them presented an approach to transform i* model into user stories, to represent an evolutionary approach for agile projects. Therefore, this paper presents an overview of a proposal to transform i* models into user stories, as an effective approach to improve the requirement engineering stages, in special, the management of requirements.

2 Objectives of the research

The main purpose of this research is to transform i* models into user stories, to facilitate the agile projects through the management of requirements. The management of requirements involves a set of activities such as documenting, modeling, prioritizing and tracking requirements, and also versioning and managing requirements changes [7]. Thus, managing requirements changes and specifying requirements, are a ways of improving and enriching the project documentation, adding, removing or editing software requirements, and relating them among the GORE approach and agile projects.

According to this proposal, requirements can be structured and prepared to be used by agile projects. Thus, the transformation proposal may assist in creating scenarios that are relevant to stakeholders and development teams. The use of an approach to assist in structuring requirements can be an alternative. However, this approach should result in consistent documents with a comprehensive overview of the proposed system, from a high level of requirements abstraction with i* to the specific functionalities of user stories.

3 Scientific contributions

The use of i* models in software projects: (i) contributes to the understanding of stakeholders intentions, and the "why" of a proposed requirements [8]; (ii) provides adequate representation of alternatives, and offers primitive modeling concepts such as softgoal and goal.[9]; (iii) is a complementary way from conventional requirements models that focuses on behaviors and interactions between systems and their environments [10].

On the other hand, user stories: (i) describe functionalities that will be valuable to either a user or purchaser of a system or software [11]; (ii) bring a clear, concise and objective description in natural language; (iii) are used to specify details and help to build tests.

Thus, the found benefits of both approaches can be useful to build scenarios during the elicitation process and requirements specification.

We argue the approach may be characterized as an evolutionary contribution where the business analysts can model the early phase of requirements stage. This result in a diagram that will become a user story to assist stakeholders in understanding their needs and in building or improving their own user stories based on results obtained with this approach.

3.1 The transformation proposal

The proposal is based in two approaches: (i) convert Strategic Dependency model (SD) in a temporary Strategic Rationale model (SR), to become understandable for the subsequent stage; (ii) isolate elements and linkages according to dependencies analysis and generate a set of user stories, represented as story cards.

We apply the semantic rules based on SR Model into SD Model to analyze the behavior of these linkages. This approach was based on Cross-Impact analysis [10], to understand how elements can be affected. Therefore, this analysis assists in understanding some semantic linkages from the i* model.
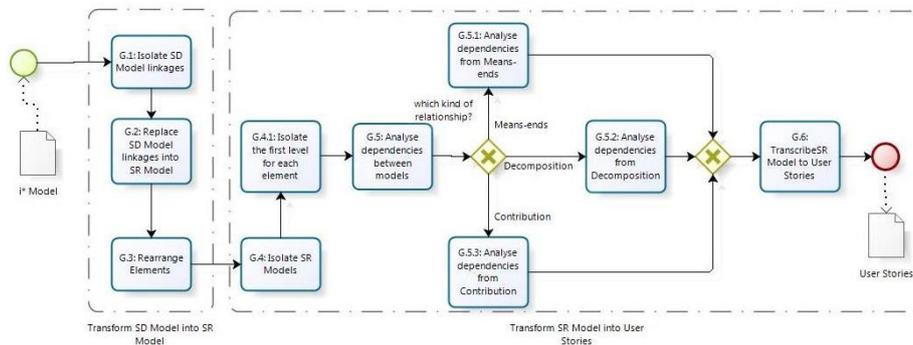

Fig. 1. Process flow using the BPMN notation.

Figure 1 describes the whole proposal process through the BPMN notation. In the first case, the transformation from SD model to SR model uses a set the guidelines applied in the i* model to perform the first phase of the process:
- G.1: Isolate the relations into small parts characterized as a requirement specification. In this case, it is necessary to analyze each dependency and their internal relations for each actor. This structure will be named as a requirement path.
- G.2: Replace the SD model linkages into SR model linkages as shown in figure 2.
- G.3: Rearrange the new SR Model vertically to become understandable when a case begins and how the case will affect the other elements.

After transforming SD model into SR model, we state that the user story should follow a set of guidelines as a way to identify dependencies:
- G.4: Isolate the SR models with their relations giving a level of relationship, i.e. only the first level of relationship among elements. The subsequent levels are not included in this user story.

- G.4.1: Consider as a one level relationship: Means-Ends, for "means" elements and their "ends"; Decomposition, for verified element and their sub-elements; Contribution, for the contributor and the contributed.
- G.5: Analyze dependencies between models, i.e., if an analyzed element is related to another isolated SR model.
- G.5.1: In a Means-Ends case, the element considered as "Ends" creates a dependency with the related "Means" element;
- G.5.2: For each Decomposition case, the element that is being decomposed creates a dependency with the element which is a part of its decomposition;
- G.5.3: For the Contribution case, the softgoal element which is contributed creates an dependency with its contributor;
- G.6: Each isolated model should be transcribed as a user story considering its dependencies among another user stories.
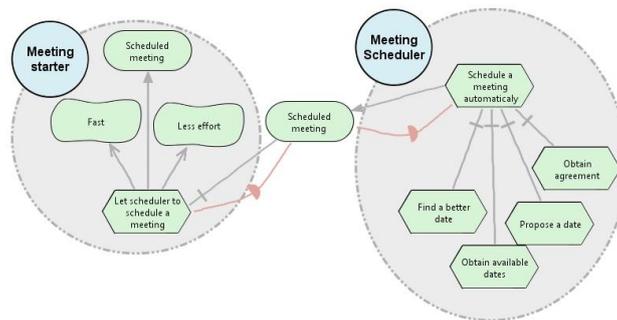

Fig. 2. Requirement path by example from [3], applying G.1 and G.2.

To illustrate the final result, table 1 shows both user stories generated after applying these guidelines.

Table 1. User Stories results

User Story 01	User Story 02
The "meeting scheduler" must "schedule a meeting automatically", to obtain a "scheduled meeting".	The "meeting starter" must "let scheduler to schedule a meeting" to obtain a "scheduled meeting".
Acceptance Criteria: • Must "find a better date"; • Must "propose a date"; • Must "obtain agreement" • Must "obtain available dates"	Acceptance Criteria: • Must obtain "scheduled meeting" by "meeting scheduler" (dependency with User Story 01) • Must achieve "fast" (treatment) • Must achieve "less effort"

4 Related works

The GORE-based transformation scenarios techniques are being used for researchers and practitioners, and some related topics have been discussed in this paper. We con-

sidered papers that use GORE modeling to be transformed into different artifacts, such as BPMN, Use Case and MDD. For each work we analyzed the problem, the solution and the result.

[12] use an approach to obtain Business Process Modeling Notation (BPMN) models from i* models and vice versa. Modeling business processes by using a goal oriented approach, such as i* framework, helps in aligning the business process improvement towards the satisfaction of the organization's strategic goals, that was not covered by BPMN. Studies about GORE modeling and BPMN were found as follows: [13] used transforming rules for i* Goal Models into Business Process Models; [14] develop an approach based on KAOS goal models to create BPMN models; [15] create a relation between BPMN 2.0 and KAOS.

[9] propose a set of guidelines to provide a mapping and transformation of Use Cases from i* models. The authors states that there's a lack of proper understanding of the organization by the software developers, and the UML is ill equipped for organizational requirement modeling, that can be represented using i* framework. This same approach can be found as follows: [5], proposes a methodology to support a co-evolution from approaches focusing on i* model, and Formal Use Cases in UML notation, that can be used in a complementary manner with requirements engineering; [16] used a KAOS approach to create diagrams in UML, in special class Diagrams.

[17] proposes a set of guidelines to generate an OO-Method (Object-Oriented) conceptual model from i* model, aiming to improve the quality of the models used on the development of information systems, and consequently to obtain better products. According to these researches, the GORE modeling transformation techniques are not common in projects based on agile methods. This lack was explored by [6] who propose a set of heuristics to perform a mapping of the requirements, presented as user stories and transformed into i* models, as a way to reduce the limitation of software context and dependencies between user stories in agile methods.

5   Conclusion, Ongoing and future work

Combining Goal-Oriented Requirements Engineering (GORE) modeling with other late-phase artifacts is becoming widely studied. This approach will contribute to reduce the gap between early-phase and late-phase as well as facilitate the management of requirements in the agile projects, by covering activities, such as documentation and elicitation. In this paper we present an overview of how to transform i* Models into user stories as an understandable document for business analysts and stakeholders and also some of the benefits related to the transformation approach of scenarios. This will help the development team to gather more information during the Requirement Engineering phase.

As future work, we will perform a research in the semantic and syntax field to propose a consistent structure, and create a friendly representation of a software requirement specification for i* models. Also, some challenges can be solved with this approach and they'll be analyzed in the future. In addition a DSL is being created to be used in a computer-based solution, as a way to transcribe i* Models into a list of requirements specification. These artifacts are planned to be used with Agile Method-

ology documents, in special User Stories. The expected impact for those issues may reflect on the academic and professional environment, since the i* model may fit in agile methods that use elicitation process for the requirements.

References

1. Robert A. Elliott, Sr., and Edward B. Allen. "A methodology for creating an IEEE standard 830-1998 software requirements specification document". In Journal of Computing Sciences in Colleges, Pages 123-131, Volume 29 Issue 2, December 2013.
2. Wiegers, K. and Beatty, J., Software Requirements. Redmond. Microsoft. 3 Ed. 2013
3. Santander, V., Integrando Modelagem Organizacional com Modelagem Funcional [Thesis]. Recife (PE). Federal University of Pernambuco. 2002.
4. Sen, A.M. and Jain, S.K., An Agile Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements Engineering. International Conference on Advanced Computing and Communications, 2007. ADCOM 2007.
5. Bhuiyan, M.M.R.; Zahidul Islam, M.M.; Krishna, A. and Ghose, A., Co-evolution of Agent Oriented Conceptual Models and Use Case Diagrams. 6th International Conference on Quality Software, QSIC 2006. 2006.
6. Jaqueira, A., Lucena, M., Aranha, E., Alencar, F. and Castro, J., Using i * Models to Enrich User Stories. Proceedings of the 6th International i* Workshop (iStar 2013). 2013
7. Pohl, K. and Rupp, C., Requirements Engineering Fundamentals. Brasil. T&M Teste de Software LTDA. 2012.
8. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J., Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems, v. 8, n. 3, p. 203-236. 2004.
9. Santander, V. and Castro, J., Deriving use cases from organizational modeling. Proceedings of IEEE Joint International Conference on Requirements Engineering, 2002.
10. Yu, E., Modelling Strategic Relationships for Process Reengineering, PhD Thesis, University of Toronto, Toronto, Canada (1995)
11. Cohn, M. and Beck, K., User Stories Applied: For Agile Software Development. Addison-Wesley Professional. 2004
12. Alves, R., Silva, C. and Castro, J. A bi-directional mapping between i* and BPMN models in the context of business process management. In: Requirements Engineering @ Brazil
13. Decreus, K., Snoeck, M. and Poels, G. Practical Challenges for Methods Transforming i* Goal Models into Business Process Models. Requirements Engineering Conference, 2009. RE '09. 17th IEEE International
14. Horita, H., Honda, K., Sei, Y., Nakagawa, H., Tahara, Y. and Ohsuga, A., Transformation approach from KAOS goal models to BPMN models using refinement patterns. In Proceedings SAC '14 Proceedings of the 29th Annual ACM Symposium on Applied Computing pp. 1023-1024. 2014.
15. Cortes-Cornax, M., Matei, A., Letier, E., Dupuy-Chessa, S. and Rieu, D., Intentional Fragments: Bridging the Gap between Organizational and Intentional Levels in Business Processes. Proceedings of Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012, Rome, Italy, September 10-14, 2012.
16. Chanvilai, S., Honda, K., Nakagawa, H., Tahara, Y. and Ohsuga, A., Goal-oriented approach to creating class diagrams with OCL constraints., in Sascha Ossowski & Paola Lecca, ed., 'SAC' , ACM, , pp. 1051-1056. 2012.
17. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J. and Pimentel, J. H., From i* Requirements Models to Conceptual Models of a Model Driven Development Process. Procdings in Second IFIP WG 8.1 Working Conference, PoEM 2009, Stockholm, Sweden, November 18-19, 2009.