A Proposed Textual Model for *i-Star*

Fabio Penha[1], Marcia Lucena[2], Leonardo Lucena[1], Celso Angra[3], Fernanda Alencar[34]

[1]Instituto Federal do Rio Grande do Norte (IFRN), Natal, Brasil
[2]Universidade Federal do Rio Grande do Norte (UFRN), Natal, Brasil
[3]Universidade de Pernambuco (UPE), Recife, Brasil
[4] Universidade Federal de Pernambuco, (UFPE), Recife, Brasil

{fabio.penha, leonardo.lucena}@ifrn.edu.br,
marciaj@dimap.ufrn.br, celso.agra@gmail.com,
fernanda.ralencar@ufpe.br

Abstract. Despite software engineering community efforts, projects continue to fail. Researches have shown that modeling became more critical due to the growth of complexity and scale of the system. In this context, the *i** Framework, a Goal-oriented Requirement Language, is used to model and analyze dependencies and intentions relationships of actors on the observed environment. However, in large-scale projects, *i** graphical notation makes the model more complex due to lack of scalability and some ambiguous representations. This paper shows a proposed textual model as a complementary vision of the *i** models. Its main objective is achieving a modeling approach that combines text and graphic to *i** Framework. We discuss how this proposal can help to mitigate scalability and complexity problems.

1 Introduction

Computational systems are present everywhere in society, playing a crucial role in various human activities. Therefore, complex social-technical systems require processes, methods, techniques and tools to capture, verify and inspect models that represent this abstracted reality. Despite the efforts of the software engineering community, project's failure rate continues an alarming growth. Therefore, the business process modeling became much more critical, since these systems expand in scale and complexity [1]. Thus, besides analyzing static and dynamic aspects, it is also necessary to consider social and intentional aspects [2]. In this context, the *i** framework is a modeling language used to develop models that represent the needs of the involved actors [2]. The *i** language has been used in several situations [2], such as telecommunications, air traffic control, agriculture, e-government, healthcare and business process. Nevertheless, it is inadequate for modeling very complex systems or systems that involve many actors [10]. The scalability of *i** is only one among its main obstacles to be used industrially [2] [10].

Some researchers have shown that graphical languages can be more intuitive than textual languages. However, both of then bring individual benefits to be evaluated when they are used together. Moreover, experience has shown that it is much harder to deal with graphic languages, especially when they need to be stricter and formal [4] [5]. Thus, we propose a textual model for the *i** Framework with the aim of making it an approach that combines textual and graphical representations, improving its advantages because we are able to achieve the following benefits through this textual model: languages integration, platform independence, versioning, formatting quality, code generators and translators [13] [6]. There is a textual model called iStarML, which differs from the proposal of this work regarding the intentionality. iStarML is an XML-based Interchange Format conceived to work with interoperability between tools and meta [14], while the textual model presented in this work is an alternative for modeling the i*, which may interest those people who are more familiar with codes or even facilitate the reading/interaction of domain expertise once it would allow us to express the system in a language closer to natural language.

This paper is structured into the following sections: Section 2 presents the construction of the textual model regarding both graphical and textual visions. Besides, some discussions about the proposal are also presented. Finally, in section 3, we present conclusions and a future work.

2 Textual Model for *i-Star*

Although the recent efforts and progress that have been made in the i* scientific community in the search for a single metamodel [8] [11], there is no consensus yet. For this reason and due to the fact of the metamodel represents only the conceptual elements, we developed the metamodel of the **Fig. 1**, based on [12], that focus on extracting the elements from the Strategic Dependency (SD) Model and the Strategic Rationale (SR) Model. The intention is to allow that other model can be granted based solely on the i* notation. This metamodel cannot fail to follow the premise that the i* framework is an actor-centered approach. Yu [2] emphasizes this concept to say that the analysis focuses on (i) how well the goals of various actors are achieved, given some configuration of relationships between human and system actors, and (ii) what reconfigurations of those relationships can help actors advance their strategic interests.

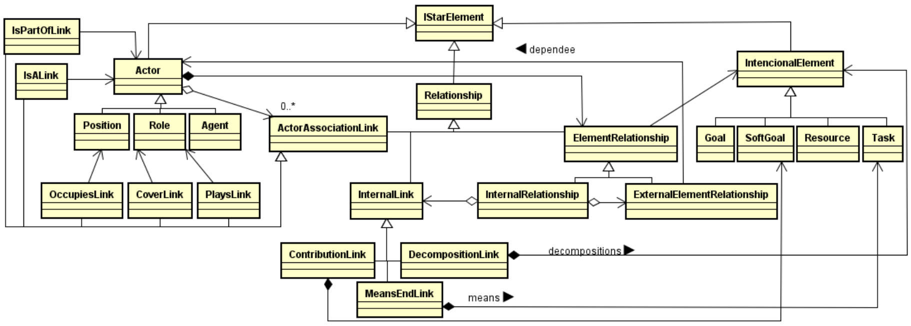
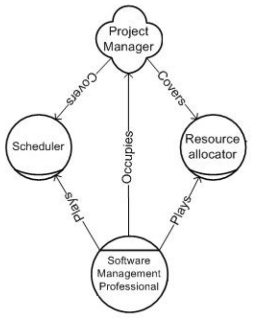
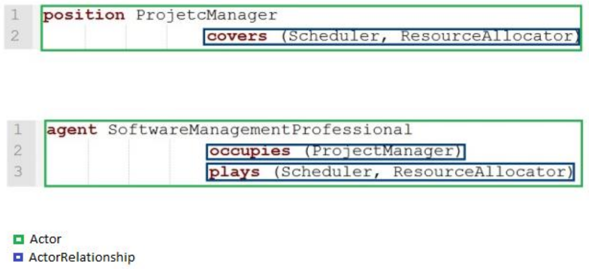


Fig. 1. The *i** Metamodel, adapted from [12]

The semantic understanding of *i** Framework enables researchers to best understand the relationship among *i** elements. This knowledge helped us depict a metamodel where the actor is framed by three elements: (i) The ***ActorAssociationLink*** defines the relationship among actors, indicating the organizational structure. (ii) The ***DependencyRelationship*** represents dependency relations of the actor through an intentional element. Semantically, there is the perspective of the *depender* and another of the dependee on the intentional element. (iii) The ***IntencionalElement*** describes an internal relationship of the actors with the intentional elements; an actor consists of intentional elements. The proposed textual model is shown through transformations from the graphical model [7] to textual model. **Fig. 2**, shows a graphic model being represented textually. It was used two actors (**Project Manager** and **Software Management Professional**) to illustrate the transformation, in which the actor may contain or not an ***ActorAssociationLink*** references other actors.



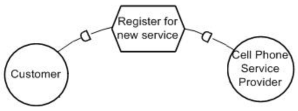


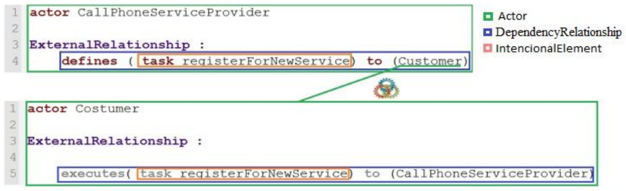
```
position ProjetcManager
                   covers (Scheduler, ResourceAllocator)
```

```
agent SoftwareManagementProfessional
                   occupies (ProjectManager)
                   plays (Scheduler, ResourceAllocator)
```

Fig. 2**.** Definition of Actor and their structural relationships (textual and graphic)

The transformation presented in **Fig. 3** describes the external relationship of an actor (**Costumer** - *depender*) that depends on another actor (**Call Phone Service Provider** - *dependee*) to perform a task (**Register For New Service** - *dependum*). This transformation shows the relationship was defined in two perspectives: the *depender* perspective and the *dependee* perspective. Note that when this dependency relationship is described in two perspectives (*depender* and *dependee*), there is a decouple in the actor modeling, i.e., everything expected to know about the actor is described in a single actor file The syntax of external relationship element complies with the intentional element (*dependum*), there is a specific syntax for external relationships for each association with a *dependum* kind (*goal*, *softgoal*, *task,* and *resource*) (**Tab 1**).





```
actor CallPhoneServiceProvider

ExternalRelationship :
    defines ( task registerForNewService) to (Customer)
```

```
actor Costumer

ExternalRelationship :

    executes( task registerForNewService) to (CallPhoneServiceProvider)
```

Fig. 3**.** Definition of an external relationship

A single file describes all external and internal relationships of the actor on modeling, which makes it possible to define modeling as a set of files corresponding to analyzed actors. **Fig. 4** and **Fig. 5** show a small transformation example of SR model (**Fig. 4**) to a textual model (**Fig. 5**). Finally, it can be observed a complete textual representation of an actor that presents us three section used in the actor's definition, which are: ***DependencyRelationship***, ***ActorAssociationLink*** and ***IntencionalElement***.

External Relationship Element (*DependencyRelationship)*		
the *Depender* perspective	***Dependum***	**the *Dependee* perspective**
defines	*Task*	**executes**
wants	*Goal*	**reaches**
wishes	*Softgoal*	**tries**
needs	*Resource*	**provides**

Table 1. Dependency Relationship Syntax

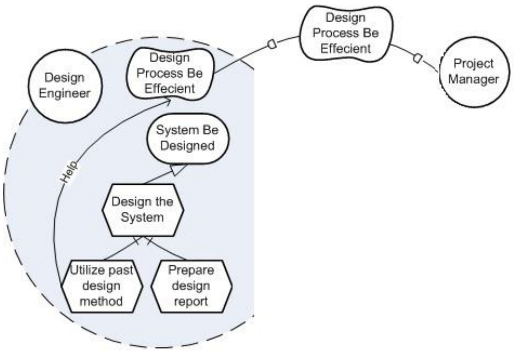
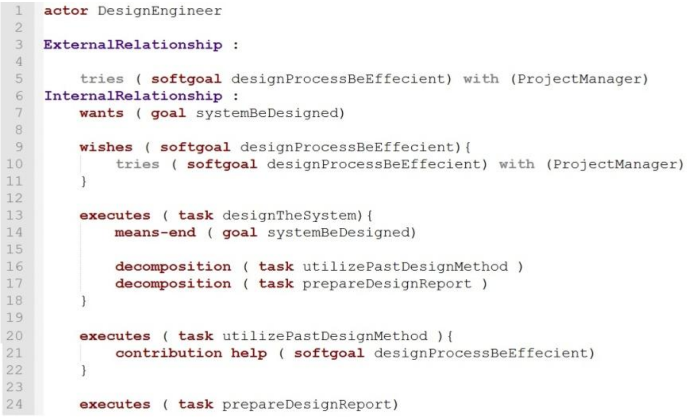


Fig. 4. The graphic model



```
actor DesignEngineer

ExternalRelationship :

    tries ( softgoal designProcessBeEffecient) with (ProjectManager)
InternalRelationship :
    wants ( goal systemBeDesigned)

    wishes ( softgoal designProcessBeEffecient){
        tries ( softgoal designProcessBeEffecient) with (ProjectManager)
    }

    executes ( task designTheSystem){
        means-end ( goal systemBeDesigned)

        decomposition ( task utilizePastDesignMethod )
        decomposition ( task prepareDesignReport )
    }

    executes ( task utilizePastDesignMethod ){
        contribution help ( softgoal designProcessBeEffecient)
    }

    executes ( task prepareDesignReport)
```

Fig. 5. single file describes all external and internal relationships of the actor

Although the proposal does not exclude the graphical models (SD and SR), it is relevant to report that the textual model can answer the following questions [2]: (i) What does each actor want? (ii) How do they achieve what they want? (iii) Who do they depend on to achieve what they want? These questions are focused on the own *i** modeling concepts. There are benefits to be considered for the proposed model. Concerning aspects of language, we have an extensible and scalable vision regarding technical aspects such as version system and compilers.

A textual language for *i** Framework is a model presenting a new perspective without changing its essence and completes the existing models. In the proposal of *iStar* 2.0 [8] have already discussed possible visions for the framework, such as the actor view. Regarding extensibility, it is possible to create mechanisms to work with textual languages, without change affecting the graphical models. E.g. using an approach to mark elements with tags, we could generate contextualized graphical models, such as @***AccountabilityDep*** to mark all relationships that involve the Accountability Department. Some concepts of scalabilities are presented in [9]: (i) the capacity to having models in different levels of abstraction, to facilitate for specialists and developers to understand the general behavior of all system or focus in a specific part of system, (ii) it is able to handle numerous Agents (actors) in an application. The textual model can be divided into a set of units, or actors, and allows looking in a specific actor at all. Besides, it provides all levels of abstractions and inserting actors without a crucial impact.

Regarding the control version system and the collaborative development, this approach uses textual files. This facilitates the integration with tools for version control, such as CVS[1], Git [2] and SVN[3] , that makes collaborative work easier. Besides, compilers can be used to automatize the processes, as following: (i) change files / actors (depender) related to an external relationship, to notify the needs to perform elements internals; (ii) transform the model into another model; (iii) improve the modeling by generating animation and simulation based on actor that will be a sequence (iv) creates mechanisms to perform a qualitative investigation of modeling, allowing an analysis that can help to reconfigure the modeling (i.e. detect if an actor is overloaded).

3 Conclusion and future work

This paper presented a metamodel composed only of the essence of the models used in the *i** framework, which allowed the creation of other proposed models without abandoning the essence. With this metamodel, a textual approach to the *i** was introduced. It was generated by transformation from graphical notation to textual one; we have also proposed a discussion about the benefits of a textual model for *i** Framework.

[1] http://www.nongnu.org/cvs/

[2] https://git-scm.com/

[3] https://subversion.apache.org/

As a future work, we are developing a toolkit to support this model with automatic mechanisms that performs qualitative and quantitative analysis modeling (i.e. compilers), as well as, mechanisms to link both approaches (textual and graphical).

4 References

1. Joseph Barjis. 2008. The importance of business process modeling in software systems design.*Sci. Comput. Program.* 71, 1 (March 2008), 73-87.
2. Eric S. Yu. 2009. Social Modeling and i*. In Conceptual Modeling: Foundations and Applications, Alexander T. Borgida, Vinay K. Chaudhri, Paolo Giorgini, and Eric S. Yu (Eds.). Lecture Notes In Computer Science, Vol. 5600. Springer-Verlag, Berlin, Heidelberg 99-121.
3. C. Nunes, J. Araújo, V. Amaral, and C. Silva, "A DOMAIN SPECIFIC LANGUAGE FOR THE I* FRAMEWORK", 11th International Conference on Enterprise Information Systems: Springer-Verlag, 04, 2009.
4. Luc Engelen and Mark van den Brand. 2010. Integrating Textual and Graphical Modelling Languages. Electron. Notes Theor. Comput. Sci. 253, 7 (September 2010), 105-120.
5. D. Harel and B. Rumpe. 2000. Modeling Languages: Syntax, Semantics and all that Stuff, Part I: the Basic Stuff. Technical Report. Weizmann Science Press of Israel, Jerusalem, Israel, Israel.
6. Mazanec, M. and Macek O. On General-purpose Textual Modeling Languages. Proc. 12th Annual International Workshop on Databases, Texts, Specifications, and Objects, Zernov, Rovensko pod Troskami, Czech Republic, pp. 1–12, 2012
7. Istarwiki.org (2016) http://istar.rwth-aachen.de/tiki-view_articles.php
8. F. Dalpiaz, X. Franch and J. Horkoff, iStar 2.0 Language Guide, dblp computer science bibliography, 2016 http://arxiv.org/abs/1605.07767v3
9. P. Lima, J. Vilela, E. Gonçalves, J. Pimentel, A. Holanda, J. Castro, F. Alencar, M. Lencastre. Scalability of iStar: a Systematic Mapping Study. CIBSE - WER. 2016
10. X. Franch. The i* framework: the way ahead. In: sixth international conference on Research Challenges in Information Science (RCIS), 2012. pp. 1–3.
11. Cares, C., Franch, X., López, L., & Marco, J.. (2010). Definition and Uses of the i* Metamodel. ISTAR.
12. M. Lucena, E. Santos, C. Silva, F. Alencar, M. Silva, J. Castro, “Towards a Unified Metamodel for i*,” RCIS 2008.
13. H. Gronniger, H. Krahn, B. Rumpe, M. Schindler, and S. Volkel. Textbased Modeling. In 4th International Workshop on Software Language Engineering, 2007.
14. Cares, C., Franch, X., Perini, A., Susi, A.: iStarML: An XML-based Model Interchange Format for i*. In: Castro, J.B., Franch, X., Perini, A., Yu, E. (eds.) Proc. 3rd Int. i* Workshop, Recife, Brazil, February 11-12, 2008, vol. 322, pp. 13–16. CEUR Workshop Proceedings, CEUR-WS.org (2008)