Using *i** and Tropos in a Software Engineering Contest: Lessons Learnt and Some Key Challenges

João Pimentel[1], Emanuel Santos[1], Bárbara Santos[1], Clarissa Borba[1], Josias Paes[1], Carlos Lima[1], André Bezerra[1], Jaelson Castro[1], Fernanda Alencar[2], Carla Silva[3], Ricardo Ramos[4], Marcia Lucena[5]

[1] Universidade Federal de Pernambuco - UFPE, Centro de Informática, Recife, Brazil
{ jhcp, ebs, bss, ccb, jpsj2, cdql, alrb, jbc }@cin.ufpe.br
[2]Universidade Federal de Pernambuco - UFPE, Departamento de Eletrônica e Sistemas, Recife, Brazil,
fernandaalenc@gmail.com
[3]Universidade Federal da Paraíba - UFPB, Centro de Ciências Aplicadas e Educação, Rio Tinto, Brazil
ctaciana@ccae.ufpb.br
[4]Universidade Federal do Vale do São Francisco – UNIVASF, Colegiado de Engenharia da Computação, Juazeiro- BA, Brazil
ricargentonramos@gmail.com
[5]Universidade Federal do Rio Grande do Norte - UFRN, Departamento de Informática e Matemática Aplicada Natal, Brazil
marciaj@dimap.ufrn.br

Abstract. In this paper we present some of the lessons learnt when using *i** and Tropos in the SCORE 2009 competition (Student Contest on Software Engineering Contest). During the development of the BTW-UFPE Project we had to address several challenges, including: limitations of modeling notation, ensuring the quality of the intentional models, transition from requirements to architecture description as well as from architecture description to detailed design. Moreover, we identified the need to deal with intentional and domain variability in *i** models and the lack of appropriate tool support. In this paper we also present some of the ongoing research which is aimed at addressing some of the identified challenges.

Keywords: *i*, Tool Support, Reuse*

1 Introduction

In the years of 2008 and 2009, we fully developed a multi-agent system which was a finalist of the Student Contest on Software Engineering – SCORE 2009 [18]. We chose the "BTW - If you go, my advice to you" project, which is related to the development of an information recommender system intended to help travelers when walking around streets unknown to them [11]. We adopted an agent oriented approach, based on the best practices of Tropos [21], to deliver our project. Agents are a natural choice when it comes to advice suggestion [24]. In the sequel we report on some of the lessons learnt and present various challenges for the technology adoption:

i) Requirements Elicitation - Our project relied on the PRiM process [7] for the elicitation phase. It was very useful for identifying actors, tasks and resources. However, it was of limited assistance for the discovery of goals and softgoals. An interesting possibility would be to use an approach based on Ground and Activity Theory [5] for finding (soft) goals. Unfortunately, due to time limitation we could not pursue this option.

ii) Different versions of the modeling language and tool support- There are different versions of the requirements language. For example there is the original *i** [22] and the *i** wiki version. Besides, different dialects have been proposed by the research groups, such as Tropos [4] and GRL [2]. In particular, in our group we have developed extensions to deal with variability and cardinality of elements [3]. We also developed two approaches to improve the modularity of the requirements models [9, 17]. This large collection of i* dialects leads to uncertainty when learning and selecting the most appropriate version for the job at hand. Moreover, the lack of standardization also constrains the usage of modeling tools, since most of them are designed for a specific version of the notation. It is urgently required a family of tools to support the various goal modeling language variants. For the Score contest we modeled requirements according to the U-Tropos process [21] and used the OME tool.

iii) Quality of models - Once we have built a model, we need to assess its quality with respect to some criteria. Several metrics have been defined for the *i** language [16, 6]. However, we also need an approach to relate the criteria of interest (quality attributes) to the questions to be answered and metrics to be collected. Moreover, the evaluation phase should also be linked to an improvement stage, where the potential problems detected could be addressed [14]. In our project we had to rely on the team members' expertise to assess the quality of the models.

iv) Transition from requirements to architecture models – A key challenge is to relate requirements and architectural models. Although some approaches have been proposed in the literature [9], the available tool support is very limited. In our project we did not use any systematic means for the derivation of the architectural model. Hence, it was difficult to assess if the derived architecture fulfilled (all) the requirements. Furthermore, the rationale for the choice of a candidate architecture (based on non-functional requirements or softgoals) was not recorded. This is especially critical on iterative projects.

v) Transition from architecture models to detailed design - Once the architectural model is stable, a detailed design has to be delivered. This is a daring task. Little assistance is currently provided. We relied on some UML artifacts to describe the design information. The members experience was a key factor for this task. A more systematic approach is urgently required.

vi) Transition from detailed design to source code – In the context of object-oriented development there are plenty of tools to generate a draft source code from design models. We predict that a draft source code could also be automatically generated from an architectural and detailed design models. For the contest our team had to generate the target code manually.

vii) Reuse of Multi Agent Systems - A lot of effort is necessary to develop a single multi-agent system (MAS), i.e. the BTW in our SCORE competition. If a similar or related MAS is required, often none of the previous artifacts are reused. Hence, it is paramount to promote software reuse in the context of multi-agent development. One of the key issues is to be able to express the common and variable parts of the artifacts. For example, there are recent works representing variability in *i** models [1, 3, 8]. However, it is not clear yet how this variability information can be used to develop an Agent-based Product Line (APL).

Our research group is addressing some of these issues, namely ii, iii, iv and vii. In the remainder of this paper we are going to describe our current research lines related to tool support and promotion of reuse. Some of the other issues are partially handled in other works [9, 14, 17]. In Section 2, we describe our research objectives. In Section 3, some contributions and published works will be discussed. In Section 4 we present some conclusions. The last section points out some ongoing and future works.

2 Research objectives

Regarding the issues describe in the earlier section, in this paper we describe our research towards the following directions:

(1) Tool Support: an SPL approach

In the last few years, several extensions of the modeling language based on the *i** / Tropos framework have been proposed, due to the specific needs of various research groups, eg. [3, 4, 17]. However, building a suitable tool support for each one of these extensions leads to a high development cost. In the mean time, the Software Product Line – SPL paradigm [13] has gained significant popularity in the software industry and academia. It promotes software reuse by specifying a family of software products through artifacts capturing their common and variable features. Thus, we aim to use their principles to provide a set products, i.e. specific goal modeling tools, to support different versions of *i**/Tropos. Each tool will be configured according to a set of specific features related to chosen modeling language [15].

(2) MAS Reuse: An SPL approach

Tropos [4] is considered one of the most complete agent oriented methodologies, since it spans all stages of multi-agent systems development. Since the initial proposal, in 2000, various versions and extensions have been proposed. However, these proposals have adopted different activities and notations, decreasing their adoption by software developers [20]. Our goal is to extend the Tropos process to enable the development of multi-agent systems according to the SPL approach. Hence, we need to add *Domain Engineering* and *Application Engineering* phases. Moreover, some form of *Feature Modeling* and *Configuration Knowledge* may also be required.

3 Contributions

In this section we describe some of the ongoing work of our research group, related to the fulfillment of the research goals presented in Section 2.

(1) Tool Support: an SPL approach

Due to the diversity of goal modeling languages based on *i**, we needed to identify the common and variable constructors present in the several *i*/Tropos* extensions/dialects. Our purpose is to develop a product line of tools that can be easily configured to support any of the analyzed extensions. Hence, inspired by the Software Product Line paradigm [13] we are defining Core Assets, as well as Domain Engineering and Application Engineering Phases. Based on common *i*/Tropos* constructors, we proposed a core metamodel to support the goal modeling variability [10]. Depending on the language we want to use, different constructors could be inserted in the core metamodel, producing a new metamodel for a specific *i** extension. As a first result, we developed a version of our *i** modeling tool - called iStarTool [15] – which currently supports the original version of *i** [22].

(2) MAS Reuse: An SPL approach

Initially we considered best practices of the several Tropos approaches. This resulted in U-Tropos: a proposal for an unified process to develop agent oriented software [21].

In the current phase of this work, the SPL technology [13] has been investigated as an alternative to promote reuse in agent oriented systems development. In this context, we examined how goal modeling languages could be used to support product line variability. In particular we tried to relate goal models to feature models [3].

Goal oriented requirements engineering (GORE) can be used to discover variable and common requirements in a software product line (SPL), as well as to reduce costs related to the configuration of a specific product in such product family. Recently, a comparison among some GORE approaches to deal with software variability has pointed out that they have limited expressivity to represent variability in SPL, as presented in [3]. This has motivated us to investigate the use of i* framework as a GORE approach for SPL. The work presented in [19] proposes an extension of the i* modeling language, called **i*-c** (*i* with cardinality*), which allows the insertion of cardinality in some of their modeling elements. The **G2SPL** (*Goals to Software Product Line*) approach proposes a process to identify and model common and variable requirements in a SPL using i*-c models. This approach also guides the configuration of a specific product in a SPL.

4 Future and Related works

In this section we present future works we plan to perform in order to achieve the goals described in Section 2. We also present some related works.

(1) Tool Support: An SPL approach

Also based on SPL concepts [13], we intend to use the core *i** metamodel, the extended *i** metamodels and their identified variabilities, to create and configure a family of tools to support goal modeling. We expect that this solution will improve the maintainability and extensibility of the current and future tools. The next product will support the *i** wiki version. Later aspectual *i** [17] will also be incorporated in the product line. We also envisage support for *i**-c (i* with cardinality)[3].

(2) MAS Reuse: An SPL approach

It is intended to extend the U-Tropos Process to include Domain and Application Engineering, for the development of agent based software using the methods and techniques of SPL. This new version of Tropos will be called Tropos-SPL (Tropos Software Product Line).

(3) Related works

We are also proposing an approach that combines variability analysis and non-functional requirements to drive the configuration of a business process. Applying this approach we can analyze variability in the model in order to assess the impact of the choices on the process quality constraints - the non-functional requirements. Moreover, it provides a rationale for the selection of a specific configuration and could support the variability representation in business process [23].

Lastly, we are using *i** models as a basis for identifying situations in which certain failures may be ignored [**¡Error! No se encuentra el origen de la referencia.**]. This work is being developed in the context of self-configurable systems, in which each failure would lead to a compensation.

References

1. Ali, R., Dalpiaz, F., Giorgini, P.: A goal modeling framework for self-contextualizable software. Book Chapter. In: Lecture Notes in Business Information Processing, vol. 29 (Enterprise, Business-Process and Information Systems Modeling), p. 326-338, 2009.
2. Amyot, D., Horkoff, J., Gross D., Mussbacher, G.: A Lightweight GRL Profile for *i** Modeling. In: Proceedings of Third International Workshop on Requirements, Intentions and Goals in Conceptual Modeling – RIGiM'09, Advances in Conceptual Modeling – Challenging Perspectives. LNCS 5833, p. 254-263, 2009.
3. Borba, C., Silva, C.: A Comparison of Goal-Oriented Approaches to Model Software Product Lines Variability. In: Proceedings of Third International Workshop on Requirements, Intentions and Goals in Conceptual Modeling – RIGiM'09, Advances in Conceptual Modeling – Challenging Perspectives. LNCS 5833, p. 244-253, 2009.
4. Castro, J., Kolp, M., Mylopoulos, J.: Towards Requirements-Driven Information Systems Engineering: The Tropos Project. In: Information Systems Journal, vol. 27, p. 365-389, 2002.
5. Cruz Neto, G., Gomes, A.S., Castro, J.: Mapping Activity Theory Diagrams into *i** Organizational Models. In: Journal of Computer Science and Technology, vol. 5, p. 57-63, 2005.
6. Franch, X.: A Method for the Definition of Metrics over *i** Models. In: Proceedings of 21st International Conference on Advanced Information Systems (CAiSE'09), 2009.

7. Grau, G., Franch, X., Maiden, N.: PRiM: An *i**-based process reengineering method for information systems specification. In: Journal of Information & Software Technology, vol. 50, p. 76-100, 2008.
8. Lapouchnian, A., Mylopoulos, J.: Modeling domain variability in requirements engineering with contexts. In: Proceedings of 28th International Conference on Conceptual Modeling (ER'09). LNCS 5829, p. 115-130, 2009.
9. Lucena, M.J.N.R., Castro, J., Silva, C.T.L.L., Alencar, F.M.R., Santos, E.B., Pimentel, J.H.C.: A Model Transformation Approach to Derive Architectural Models from Goal-Oriented Requirements Models. In: Proceedings of 8th International Workshop on System/Software Architectures (IWSSA'09). LNCS 5872, p. 370-380, 2009.
10. Lucena, M.J.N.R., Santos, E., Silva, M.J., Silva, C.T.L.L., Alencar, F.M.R., Castro, J.F.B.: Towards a Unified Metamodel for *i**. In: Proceedings of Second International Conference on Research Challenges in Information Science (RCIS'08), 2008.
11. Pimentel, J.; Santos, E.; Castro, J. Conditions for ignoring failures based on a requirements model. In: Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering (SEKE), 2010, in press.
12. Pimentel, J.H.C., Borba, C., Xavier, L.: BTW: if you go, my advice to you Project. July, 2009. Web-site. In: https://jaqueira.cin.ufpe.br/jhcp/docs/ (last access, Feb 2010).
13. Pohl, K., Bockle G., Linden, F.V.: Software product line engineering. Springer: Verlag, Berlin, Heidelberg, 2005.
14. Ramos, R.A.: AIRDoc - An Approach to Improve the Quality of Requirements Documents: Dealing with Use Case Models. PhD Thesis. Federal University of Pernambuco, 2009.
15. Santos, B. S., Paes Junior, J.: iStarTool. 2009. Web-site. In: http://portal.cin.ufpe.br/ler/Projects/IStarTool.aspx (last access, Feb 2010).
16. Santos, E. B.: A Proposal of Metrics to evaluate *i** Models (in Portuguese: Uma Proposta de Métricas para Avaliar Modelos *i**). MSc Dissertation. Federal University of Pernambuco, 2008.
17. Alencar, F., Castro, J., Lucena, M., Santos, E., Silva, C., Araújo, J., Moreira, A.: Towards Modular *i** Models. In: Proceedings of Requirements Engineering Trank, 3th Ed., 25th ACM Symposium on Applied Computing, 2010.
18. SCORE 2009. Web-site. http://score.elet.polimi.it (last access, Jan 2010).
19. Silva, C., Borba, C., Castro, J.: G2SPL: A Goal Oriented Requirements Engineering Process for Software Product Line (In Portuguese: G2SPL: Um Processo de Engenharia de Requisitos Orientada a Objetivos para Linhas de Produtos de Software). In: Proceedings of 13th Workshop on Requirements Engineering (WER), 2010 (to appear).
20. Silva, M.J., Maciel, P., Pinto, R., Alencar, F., Tedesco, P., Castro, J.: Extracting the Best Features of Two Tropos Approaches for the Efficient Design of MAS. In: Proceedings of X Iberoamerican Workshop on Requirements Engineering and Software Environment (IDEAS'07), p. 3-16, 2007.
21. Silva, MJ.: U-TROPOS: an unified process approach for agent oriented software development (In Portuguese: U-TROPOS: uma proposta de processo unificado para apoiar o desenvolvimento de software orientado a agentes). Msc dissertation. Federal University of Pernambuco, 2008.
22. Yu, E.: Modelling Strategic Relationships for Process Reengineering. PhD thesis. University of Toronto, 1995.
23. Santos, E. B., Pimentel, J. H., Castro, J., Sanchez, J.: Configuring the Variability of Business Process Models Using Non-Functional Requirements. In: Proceedings of 15th International Conference on Exploring Modelling Methods for Systems Analysis and Design (EMMSAD'10).
24. Montaner, M., López, B., De la Rosa, J. A Taxonomy of Recommender Agents on the Internet. In Artificial Intelligence Review, June 2003.