10326 - A STUDY OF THE SIRA PROCESS
(SYSTEMATIC INTEGRATION BETWEEN REQUIREMENTS AND ARCHITECTURE)

Turah Xavier de Almeida¹; Jaelson Freire Brelaz de Castro²

¹Estudante do Curso de Engenharia da Computação – CIn– UFPE; txa@cin.ufpe.br, ²Docente/Pesquisador do Centro de Informática – CIn – UFPE; jbc@cin.ufpe.br

Abstract: Requirements Engineering and Software Architecture are recognized within the software engineering community as important areas of research and practice. Even though requirements and architecture are clearly related, the transition between them remains as a challenging problem since there is still a lack of systematic guidelines for this transition. The SIRA Process focuses on a systematic way to assist the transition from requirements to architecture. The aim of this work is to evaluate the SIRA process in order to determine if it is possible to develop a CASE tool able to support the SIRA process automatically.

Key-Words: requirements engineering; software architecture; multiagent systems; tropos; sira

INTRODUCTION

Both Requirements Engineering and Software Architecture are recognized within the software engineering community as important areas of research and practice. During software development, there is a large conceptual distance between what to do (requirements) and how to do it (architecture, design and code). Even though requirements and architecture are clearly related, it is surprising how little research has been done so far to systematically derive architecture based on requirements. Currently, this process is a difficult task mainly based on intuition and experience of architects and designers. The transition between requirements and architecture remains as a challenging problem since there is still a lack of systematic guidelines to support this transition.

A lot of effort has been made to try and solve this problem. The following processes can be highlighted: Architecture Description Languages (ADL), Goal-Based Approach, Problem Frames, Use Case Maps (UCMs), Rule-Based Decision Making, Architecting Requirements and Object-Oriented Transition [6].

It is in this context that the SIRA Process comes into place: the SIRA Process focuses on a systematic way to assist the transition from requirements to architecture. It is provided by the SIRA Framework that was presented in [3] as a framework able to help reduce the gap among Multiagent Systems (MAS) requirement models and architectural models. This framework describes a software system from the perspective of an organization and is located in the context of the Tropos Project [4].

The aim of this work is to evaluate the SIRA process in order to determine if it is possible to develop a CASE tool able to support the SIRA process automatically. A case study will be applied in each activity of the process to try to resolve its conflicting decisions. Mechanisms will be created to help provide resolutions and explanations when multiple conflicting decisions are made for the same part of the SIRA model, such as goal refinement, cluster analysis and correlation analysis [3]. Possible failures in the process will be identified and corrected (when possible) and suggestions for improvement will be made.
METHODOLOGY

The methodology adopted by this work is decomposed in phases (each phase has a goal to be achieved) which will be presented next: (i) **Study of the concepts of agent oriented modeling.** The aim of this phase is to understand the agent oriented modeling as software development; (ii) **Study Organizational Modeling concepts (i*, TROPOS).** The aim of this phase is to understand the concepts of i* modeling and how it is applied in modeling systems; (iii) **Study of the SIRA Process.** The aim of this phase is to understand the SIRA process. In order to do so, papers related to the SIRA process as well as the transition between requirements and architecture will be studied, especially [3] and [6]; (v) **Evaluation of the SIRA Process.** The aim of the last phase is to determine if it is possible to develop a CASE tool able to support the SIRA process automatically. In order to do so, a case study will be applied in each activity of the process to try to resolve its conflicting decisions. Mechanisms will be created to help provide resolutions and explanations when multiple conflicting decisions are made for the same part of the SIRA model and possible failures in the process will be identified and corrected (when possible) and suggestions for improvement will be made.

RESULTS AND DISCUSSIONS

In this work, the Conference Management case study was applied to each activity of the process in order to evaluate it and determine if it is possible to develop a CASE tool to support it automatically. We came to the following results: (a) In **Goal and Task Refinement** sub-activity, it is not clear to what extent and when goals and tasks should be refined. We came up with the following solution: the developer must refine goals and tasks until enough information is available for the next step of the process, the **Identifying Roles** step. In the Conference Management example, further refinement was not necessary as roles could be identified from the current refinement of goals; (b) In **Role Identification** sub-activity, it is not clear how to identify the roles and how to create the role interaction graph. (c) In **Cluster Analysis** sub-activity, it is not explained how to proceed when roles do not communicate with other roles and therefore, their farness values can not be calculated. Future work is required to deal with this issue as this work could not come up with a solution for that. In the Conference Management example, this missing information was not relevant and there was no problem in carrying on with the process; (d) In **Correlation Analysis** sub-activity, it is not explained how to combine the similarity matrix based on in-degree and the similarity matrix based on out-degree to generate the similarity correlation table. The solution given for the Conference Management example was based on how the NFR framework deals with conflicts (negative correlations) and harmony (positive correlations) of non-functional requirements in software engineering; (e) In **Architectural Configuration** activity, once again, it is not explained how to combine the similarity correlation table and the centrality correlation table to generate the correlation analysis table. The solution given for the Conference Management example was based on how the NFR framework deals with conflicts (negative correlations) and harmony (positive correlations) of non-functional requirements in software engineering.

During this work, some difficulties were faced, such as: It was not clear in [3] whether the activities of the process required any human interaction or could be performed by an algorithm. In this work, we came to the conclusion that **goal and task refinement** and **role identification** sub-activities are mainly based on human domain knowledge, experience and intuition and therefore, can not be automatized.

CONCLUSIONS

Understanding the relationship between software requirements and architecture remain as a challenging software engineering problem. The SIRA Process focuses on a systematic way to
assist the transition between requirements and architecture. As there is currently a lack of systematic guidelines to help this transition, the purpose of the SIRA Process is very relevant to software engineering. Its main idea is to systematically derive the architecture of a multi-agent system by analyzing its requirements based on organizational concepts. Despite its relevance, some aspects of the SIRA Process require further work. Considering this, future work could take the following directions: (a) Consider the semantic aspects of a model when grouping roles. To do so, it might be necessary to have even more human participation; (b) Study Goal and Task Refinement and Role Identification sub-activities in order to determine the best way in which they should be carried out as they can not be automatized; (c) Resolve the conflicting decisions that could not be resolved by this work, such as determining farness values when roles can not communicate with all other roles; (d) Apply the SIRA Process in many different applications to obtain more information about its weaknesses and strengths; (e) Develop tools to guide the Architectural Selection, Cluster Analysis and Correlation Analysis sub-activities, as well as the Architectural Configuration activity, as they can be automatized.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Jaelson Castro, for taking me into his group and for all the support and guidance I have been given. Also, I would like to thank Márcia Lucena for her guidance, enriching discussions over this work and for being available to help me whenever I needed. Este trabalho é incentivado e financiado pelas concessões de pesquisa: CAPES/GRICES Proc. 129/05; SOFTAS Project, POSC/EIA/60189/2004.

REFERENCES