IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 317-320

An Efficient Framework for Requirement Engineering

Chander Diwaker[1], Kavita[2], Ajay Jangra[3] & Shikha[4]

[1,2,4]CSE Deptt. U.I.E.T., Kurukshetra University, Kurukshetra
[3]CSE, Deptt. Guru Jambheshwar University (hr.), India
Email: [1]chander_cd@rediffmail.com, [2]Kavitadawra285@gmail.com, [3]er_jangra@yahoo.co.in, [4]shikha.mtech28@gmail.com

ABSTRACT

Software engineering is an engineering discipline and use of sound engineering principles that concerned with the theories, method and tools with all aspect of software production. Requirement engineering is a branch of software engineering concerned with the real world goals for function and constraints on software systems. Requirement engineering is a process that establishes the foundation on how the software system has to be implemented and related with other processes of software life cycle. In this paper we propose an efficient framework with tropos methodology, based on agent oriented framework for the software product. Requirement engineering (RE) process technique in the project is the first step towards increasing the overall quality of the software product. This paper also presents a framework that guides the requirements engineer to model the requirements engineering process in an efficient manner.

Keywords: Software Development, RE Process Framework, Requirements Engineering Methodology and Process, Goal Oriented Requirement Engineering, Soft and Hard Goals.

1. INTRODUCTION

Software engineering process is concerned with set of activities, method, practice and transformation that people use to develop and maintain software product. The overall objective of RE is to build a sound requirement specification for the system to be developed. Currently in requirement engineering the attention is being more and more on the understanding of the problem by studying in existing organizational setting in which the system will operate.

Requirement engineering is a disciplined approach that concerned with analyzing and documenting requirements. Requirement engineering process is of establishing the services that the customer requires from a system and constraints under which it operates and developed. This approach is to managing task and responsibilities in a development organization. The process involve in requirement engineering includes eliciting, understanding, analyzing and documenting requirements. [1, 2]

Requirement engineering(RE) is a discipline that provide a set of process to identify and engineer requirement, not only manage but also analyze, validate and traceable. The main strategy of requirement engineering on how to engineer a requirement rather than how to manage the requirements. This framework is simple, effective and represents the graphical notation that improves the quality of the system. Tropos deals with all the phases of system requirement analysis and all the phases of system design and implementation in a uniform and homogeneous way, In order to ensure the quality of a software requirements specification, there needs to be emphasis on implementing engineering disciplines into the RE process by using various techniques and methodologies.

2. BACKGROUNDS AND RELATED WORK

A requirement is a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. A well-formed requirement is a statement of system functionality (a capability) that must be met or possessed by a system to satisfy a customer's need or to achieve a customer's objective, and that is qualified by measurable conditions and bounded by constraints. The processes involved in RE include domain analysis, elicitation, specification, assessment, negotiation, documentation, and evolution. Getting high quality requirements is difficult and critical. Recent surveys have confirmed the growing recognition of RE as an area of utmost importance in software engineering research and practice.

In this paper, we propose a framework to characterize typical requirements engineering techniques and use it as a base for selecting appropriate techniques at the time of starting a project as well as at the time of recognizing a change in the project nature or encountering an obstacle in defining a suitable set of requirements. Tropos, a novel agent-oriented software engineering methodology, is heavily characterized, among other features, by the fact

that it pays great attention to the activities that precede the speciation of the prescriptive requirements, such as understanding how the intended system would meet the organizational goals.The focus here is on the early requirements and on how to manage the transition from them to the late requirement analysis. A goal-oriented requirements engineering methodology aiming at modeling not only what and how aspect of requirements but also why, who, and when. [3, 6, 7]

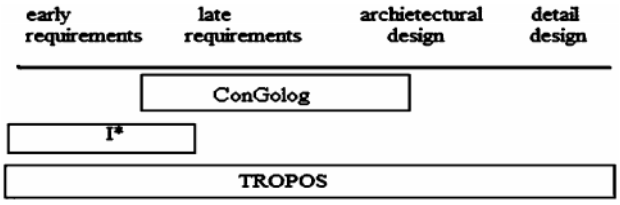
Requirements engineering still remains a key factor to guarantee stakeholders involvement, facilitating their understanding and participation. This paper introduces *REF*, an agent-based Requirements Engineering Framework designed around the adoption of a simple, but effective, representational graphical notation. [5]

3. COMPARING GOAL ORIENTED METHODOLOGIES

Tropos is a software development methodology founded on the key concepts of agent-oriented software development. Specifically, *Tropos* emphasizes concepts for modeling and analysis during the early requirements phase. The Tropos methodology addresses agent-oriented development and it is intended to support all analysis and design activities in the software development process. One of the main advantages of the Tropos methodology is that it allows to capture not only what or the how, but also the why a piece of software is developed. The 'why' question helps to discover the objectives and rationale behind the goals which in fact identify the higher goals. It also helps to improve the quality. Tropos supports five phases of software development. The early requirements analysis is concerned with the understanding of a problem by studying an existing organizational setting. [8, 9, 10, 18]

During the late requirements analysis, the system-to-be is described within its operational environment, along with relevant functions and qualities. The architectural design phase deals with the definition of the system global architecture in terms of subsystems that are represented as actors, and their data dependencies, that are represented as actor dependencies. The detailed design phase aims at specifying each architectural component in further detail in terms of inputs, outputs, control and other relevant information. Finally, during the implementation phase, the actual implementation of the system is carried out, consistently with the detailed design. Tropos methodology itself, called, since now on, the Tropos tool. [8, 11, 18]

The **i*** framework [Yu95, Yu97] was developed for modeling and analyzing organizations to help support business process reengineering and requirements engineering. The framework focuses on modeling intentional and strategic relationships between actors.[12] The *i** framework is a goal-oriented and agent-oriented language defined with the aim of modeling and reasoning about organizational environments and their Information Systems.



The *i** framework does not address the construction of the *i** models as a main issue, but it provides some guidance on how to develop them based in constructing the SR(Strategic Rationale) models by using strategic reasoning in order to obtain also the SD (Strategic dependencies).

Re Framework Features	*Tropos*	*I**	*ConGolog*
Ease of use	Formal	Diagram based informal	Expressive logic based
concept	Agent, situation, axioms	Actor, goal, task, resources	I* concept with specification language
RE activities	Specifying and analyzing early require-ment	Early phase of RE	Late phase of RE
Actor participation	Design choice autonomy	Design choice autonomy	Formal lang of agent programming
Tool support	T -tool	No	Simulation v & v tool
Coverage of life cycle	Early design, implementation	Early, late requirements	Early analysis, design

Fig.1: Comparison between Different Methodologies with Different Features

In the i* framework [Yu93, Yu97], various types of agent dependency links are defined to model situations where an agent depends on another for a goal to be achieved, a task to be achieved, or a resource to become available.

ConGolog is a framework for process modeling and agent programming. It is based on the situational calculus, a language of predicate logic for representing dynamically changing worlds. The ConGolog framework can be used to model complex processes involving loops, concurrency, multiple agent etc. Because it is logic-based framework that can accommodate incompletely specified models, either in the sense that the initial state of the system is not completely specified, or that the processes involved are nondeterministic and may evolve in any number of ways. A ConGolog model of a domain includes two components. One is the specification of the domain dynamics, i.e. how to model the state, what is

the initial state of the domain, what actions can be performed, when the actions can be performed and what their effects are. Another is the specification of the process of interest, i.e. the behavior of the agents involved in the domain. So Tropos covers the software development process as a whole, from the early steps, in which the software engineer picks up and models requirements of the organizational setting (early requirements) and of the system-to-be (late requirements), up to the detailed design. [13]

4. THE PROPOSED FRAMEWORK

Research methodology provide the scientific way of solution for the given problem and analysis of the problem. Tropos proposes a software development methodology and a development framework which are founded on concepts used to model early requirements and complements proposals for agent-oriented programming platforms. In this paper we present a general framework, which is useful during agent-oriented development. *Tropos* [Trop] is an agent-oriented software development methodology which is founded on the concepts of goal-based requirements adopted from the i* [I Star] and GRL (Goal-oriented Requirements Language) [GRL].Tropos supports five phases of software development. Early requirements analysis is one of the most important and difficult phases of the software development process. It is the phase where the requirements engineer is concerned with understanding the organizational context for an information system, and the goals and social dependencies of its stakeholders. Late requirements analysis results in a requirements specification, which describes all functional and non-functional requirements for the system-to-be.

This paper presents requirements engineering framework based on the notions of *Actor*, *Goal*, and *Intentional Dependency*. The framework provides the analyst with a powerful tool for capturing high-level organizational needs and transforming them into system requirements in a smooth and controlled manner.

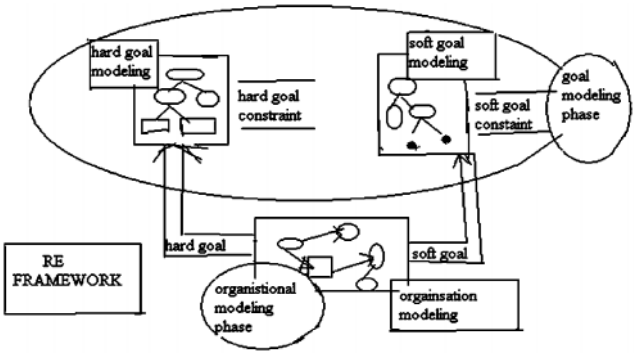


Fig. 2: Proposed Requirement Engineering Framework

This framework describes the three types of modeling in requirement engineering process model:

Organization Modeling deals with the application domain structure. The key modeling abstraction is the Agent in which the organizational context is analyzed and the agents and their goals identified.

Hard-Goal Modeling determines how to achieve hard goals, by decomposing them into sub hard goals and tasks. This is used to determine how an agent can achieve a received hard goal, by decomposing it into more elementary subordinate hard-goals, *tasks*, and *resources*. *Constraints* may be associated with hard-goals, tasks, and resources to specify the corresponding quality attributes. [4]

Soft-Goal Modeling produces operational definitions of the soft goals, to capture and formalize the users' semantics. This producing the operational definitions of the soft-goals, sufficient to capture and make explicit the semantics that are usually assigned implicitly by the involved agents. The main aim during Soft- Goal Modeling is to iteratively refine each soft-goal in terms of subordinate elements, until only hard-goals, tasks, resources, and *constraints* are obtained or each not refined soft-goal is passed on another agent, in the context of which will then be refined.[2, 4, 11]

Thus, the resulting set of constraints represents the final and operationalized views of the involved quality attributes, i.e., the quality models that formalize the attributes for the specific context.

Quality Improvement

The processes involved in RE include domain analysis, elicitation, specification, assessment, negotiation, documentation, and evolution. Getting high quality requirements is difficult and critical. To acquire high quality requirement engineering, goal oriented approach was chosen. Goal-Oriented requirements engineering is an approach about the process and product, as well as functionality into a requirements object. [14, 17]

In software-intensive systems, the achievement of qualities—such as performance, availability, security, and modifiability—is dependent on the software architecture. Quality should be measured from the early stages of software building or else the development can end up with software that fulfills the requirements but fails to satisfy the customer. High Quality Requirement documents are a must for successful software projects. All process phases directly and indirectly depend on the requirement document. As Tropos methodology is that it allows to capture not only *what* or the *how*, but also the *why* a piece of software is developed. The *'why'* question helps to discover the objectives and rationale behind the goals which in fact identify the higher goals. It also helps to improve the quality. [7, 15, 16, 17]

5. CONCLUSION

In this paper, we doesn't focus on new RE technique but developing a framework which provide support for selection of existing RE process model, technique for the software product. This framework relates with the tropos methodology that is based on agent and goal oriented techniques. In this, we presented the overview of requirement engineering and introduced an agent-oriented Requirements Engineering Framework (REF), explicitly designed to support the analysts in transforming high-level organizational needs into system requirements, while redesigning the organizational structure itself. This new approach helps to improve the quality. Tropos methodology focused on its application in developing specific classes of applications, as for instance distributed knowledge management systems.

REFERENCES

[1] A. I. Ant´on, "Goal-based Requirements Analysis", *In Proceedings of the IEEE International Conference on Requirements Engineering (ICRE '96)*, Colorado Springs, USA, Apr. 2006.

[2] Axel van Lamsweerde, "Requirements Engineering in the Year 00: A Research Perspective*", 0-7695-2579-2/00 IEEE.*

[3] Hofmann, H. F. and Lehner, F. : "Requirements Engineering as a Success Factor in Software Projects", **18**, *No. 4, pp.58-66/01 IEEE*.

[4] A. Van Lamsweerde, "Goal-oriented Requirements Engineering: A Guided Tour", In *Proceedings of RE'01 - International Joint Conference on Requirements Engineering*, Pages 249–263, Toronto, Aug 2001. IEEE.

[5] Decker, B. et al, "Wiki-based Stakeholder Participation in Requirements Engineering", *IEEE Software*, March/ April 2007.

[6] Gans, G., Jarke, M., Kethers, S., and Lakemeyer, G., "Modeling the Impact of Agent Networks", to Appear in *Proc. of the 3rd International Bi-Conference Workshop* Switzerland, June, 2001.

[*7*] D. Bertolini, A. Novikau, A. Susi, and A. Perini, "TAOM4E: an Eclipse Ready Tool for Agent-Oriented Modeling", *Issue on the Development Process*, pp.45-48, 2006.

[8] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "Tropos: An Agent-Oriented Software Development Methodology", **8(3)**, *203–236, July 2004.*

[9] Paolo Bresciani1 and Paolo Donzelli2, "REF: a Practical Agent-Based Requirement Engineering Framework", *978-0-7695-3316-2/05 IEEE.*

[10] M.-P. G. Federico Bergenti and F. Zambonelli, Editors, *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, Springer, 2004, pg 121-130.

[11] Wang,X., Lesperance,Y., "Agent Oriented Requirements Engineering using ConGolog and **i*",** *Proceedings of 3rd International Bi Conference Workshop on Agent Oriented Information Systems*, AOIS01, Montreal, Canada, May2000.

[12] Anwer, S., Jinnah, M.A., Ikram, N.: "Goal Oriented Requirement Engineering: A Critical Study of Techniques", In *Proceedings of the 13th* APSEC 2006.

[14] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "TROPOS: an Agent-Oriented Software Development Methodology", Page 23-29, March 2002.

[15] Shahzad Anwer, Naveed Ikram, "Goal Oriented Requirement Engineering: A Critical Study of Techniques*", 0-8186-4300 5/06 IEEE.*

[16] Boehm, B. W., "Characteristics of Software Quality", New York,: American Elsevier, 2008.

[17] Barbacci, M.; Klein, M.; & Weinstock, C, "Principles for Evaluating the Quality Attributes of Software Architecture", *PA: Software Engineering Institute*, Carnegie Mellon University, 2006.

[18] Fuxman, A., Pistore, A., Mylopoulos, J., Traverso, P., "Model Checking Early Requirements Specifications in Tropos", *Submitted to the Fifth IEEE International Symposium RE01*, August, 2001, Toronto, Canada, available at http://www.cs.toronto.edu/km/tropos/.