

A Comparative Analysis of RE Approaches for MAS

Anuja Soni
Department of Computer Science
University of Delhi
Delhi, India

ABSTRACT

Poor requirements are one of the principal reasons for collapse of software projects. Detecting the requirements errors at the final stage intensifies the efforts of developers that may delay system delivery and overrun cost. Therefore to reduce the development time and cost, it necessitates a systematic and structured Requirement Engineering (RE) approach to establish a comprehensive understanding of the requirements addressed by a software product. Agent-based technology for software development has been recognized as a promising and pioneering paradigm for next generation systems. It was observed that a number of RE approaches in literature, are the rich sources of techniques facilitating various phases of RE. This work presents a year-wise comparative analysis of these RE approaches that could provide systematic guidelines to developers for an effective application of RE to Multi-Agent System (MAS).

Keywords

User Story Card (USC); Agent Card (AC); Multi-agent System (MAS), Validation

1. INTRODUCTION

Most software projects fail due to weak Requirement Engineering (RE) [19]. The research shows that 60% to 80% failures of the software projects are attributed to the poor requirements [16, 17]. Problems such as inadequate requirements elicitation, poor requirements analysis, weak negotiation processes and lack of appropriate requirements traceability practices hinder the realization of effective software systems.

RE is perceived as an area of growing importance in software development that signifies a systematic handling of requirements to mitigate the vulnerability of the final system caused by software flaws [14, 15]. The negligence to the users' requirements in the early phases of software development results in high development cost and longer cycle time. To mitigate these problems, a systematic and structured RE approach is essential that is equipped with the potential to establish a common understanding of the requirements addressed by the final product.

The organization of the paper is as follows: Section II presents various approaches for RE. Section III describes MAS, section IV provides a comparative analysis of various RE approaches for MAS and finally section V concludes the paper.

2. VARIOUS APPROACHES FOR REQUIREMENTS ENGINEERING (RE)

Various definitions have been introduced to define RE. Nuseibeh et al. [20] describe RE as a concern that interprets and comprehends stakeholders' terminology, concepts, viewpoints and goals, whereas Cysneiros et al. [21] perceive RE as an area that focuses on identifying and characterizing "what" the system should do. Jalote [22] defines RE as a means to produce Software Requirements Specification (SRS), which accurately captures the client's requirements and forms the basis of software development and validation. Pandey et al. [23] defines

RE as a systematic approach through which the software engineer collects requirements from different sources and implements them through the software development process. Nguyen et al. [24] describe RE as an iterative process of discrete chronological activities viz. elicitation, definition, negotiation, prioritization, validation and specification for delineating and articulating the imprecise, incomplete and the potential needs of the users into complete, precise and formal specifications. ISO/IEC/IEEE standard 29148-2011 defines RE as an interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system.

Essentially, these definitions focus on the necessity of the RE approach for meeting the multi-faceted objectives of developing a high quality software that includes bridging the gap between two parties namely users and the developers; achieving the goals of the final system by satisfying all the stakeholders, and articulating the potential needs of the users into complete, precise and formal specifications.

RE is considered as the most critical and fundamental activity for development of the software systems. A number of approaches for RE viz. Object-Oriented [25], Goal-Oriented [26], Aspect-Oriented [28], Agile [27] and Agent-Oriented [29] are reported in literature.

The Object-Oriented RE [25] interprets the requirements of a system in terms of objects that encapsulate the information about the process and product, as well as functionality into single periphery and defends the data from accidental modification.

The Goal-Oriented RE [26] is concerned with modeling and transformation of the organizational goals into the system requirements. Organizational goals may be defined as high-level objectives of a business or system at various levels within an enterprise.

The Aspect-Oriented RE [28] focuses on identification, specification and representation of crosscutting concerns of requirements and their modularization in the initial phase of software development. Crosscutting concerns are those aspects of a system that can't be clearly separated from the rest of the system and produce tangled and scrambled requirements that are difficult to maintain and evolve.

Agile RE [27] involves continuous communication and collaboration of developers with customers to address the requirements that evolve over time, prioritize, and deliver the most valuable functionalities. It is an effective way of responding to changes in requirements that results in shorter cycle time and is useful especially in situations where timing is a critical competitive advantage for an organization.

Agent-Oriented RE [29] describes the requirements of a system in terms of autonomous interacting agents. Software agents are computer programs that act autonomously on behalf of their users across open and distributed environments to solve a growing number of complex problems. Characteristics such as autonomy, reactivity, and pro-activeness as well as social ability differentiate software agents from traditional software.

Autonomy refers to the agent's ability to work on its own without the direct interventions of humans or others; reactivity is the agent's ability to perceive its environment and respond to stimulus; pro-activeness is the agents' ability to take initiatives in its environment to achieve the goals; and social ability is the agents' ability to share information with peers. Examples of software agents include artificial players or actors in computer games and simulations, trading and negotiating agents, web spiders used by a search engine for collecting data on WWW and computer viruses called as destructive agents.

The conventional modeling techniques incorporate the methods and objects which are passive and invoked only when some external entity sends them a message, while software agents have their own thread of control, restricting not only code and state but their invocation as well. The agents may have individual rules and goals, making them appear like active objects [7].

3. MULTI-AGENT SYSTEM (MAS)

As economies are becoming extremely knowledge intensive, need for effective coordination of collaborative efforts is being increasingly felt. Even in an enterprise-wide context, the need for cooperative activities needs hardly any exaggeration and similar is the case in software development industry. In software engineering community, there is an ever increasing demand to amend and streamline software projects in terms of MAS [16, 17]. MAS is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solve.

MAS is designed for a variety of domains, including, maintaining cellular switching systems, representing buyers or sellers in e-commerce transactions, automating bid-evaluation process, an intelligent wheelchair, casting vote, electrical plant condition monitoring, the elevator system, monitoring complex chemical processes, servicing mobile manipulator robot, automating matchmaking in a stable two-sided B2B market, composing web services for adaptable business processes and etc.

4. A COMPARATIVE ANALYSIS OF RE APPROACHES FOR MAS

Software agents are considered as a next generation model for engineering complex, heterogeneous and open distributed systems. Software engineering community is increasingly revising and restructuring projects in terms of MAS as it is a recent paradigm for conceptualizing, designing, and implementing software systems. Agent-Oriented RE models the requirements of a system in terms of autonomous interactive component agents [14, 15, 16, 17].

Researchers have tried various ways to address the requirements of agents. In literature various Agent-Oriented requirements frameworks such as i^* [1], Gaia [2], MaSE [3], AGORA [4], RE for large scale MAS [5], TROPOS [6], Formal Tropos [7], Enhanced model [8], RE from Craft to Discipline [9], NorMAS-RE [10], B-Tropos [12], RE4Gaia [11], and Iterative REF [13] are presented. A brief description of these is prescribed in order.

i^* [1] is well known being one of the most cited methodologies that focuses on intentional and social aspects of agents in a collaborative environment. It represents the primitive requirements of agents in terms of goals, tasks and inter-actor dependencies using the strategic dependency and rationale model diagrams.

Gaia [2] represents MAS as an organization of agent-roles using role model that is described using responsibilities, permissions,

activities and protocols. Responsibilities, permissions and activities define the functionalities, rights and private actions of agents in an open and distributed environment, whereas protocols are specified as an institutionalized pattern of interaction among agents.

MaSE [3] methodology takes an initial system specification and produces a set of formal design documents in a graphically based style. As the part of requirements analysis it facilitates capturing of goals using use cases and refinement of roles using graphical tree like structure, sequence diagrams and role models. The tree like structure facilitates the traceability of requirements by switching from one node to another.

AGORA [4] is purely a goal oriented approach that assists an analyst to choose and adopt a goal from the alternatives available, while resolving the conflicts of the stakeholders and analyzing the impact of changes in requirements using goal graph and preference matrix.

RE for large scale MAS [5] presents a methodology based on agent concepts at the requirements modeling level. By taking into account agent characteristics such as autonomy, intentionality, and sociality starting from the requirements level, the methodology leads naturally into the development of large-scale MAS.

Tropos [6] augments various mentalistic notions of agents such as actor, goal, plan, belief, resource, dependency and capabilities with the basic constructs of i^* to extend the requirements modeling from early to late phases. This methodology focuses on the comprehensive requirements analysis of various agents using Means end analysis, Contribution analysis and AND-OR decomposition of goals in several modeling diagrams that produce a complete set of system requirements involving both functional and non-functional.

Formal Tropos [7] is based on a specification language that adopts primitive concepts for modeling early requirements (such as actor, goal, and strategic dependency), along with a rich temporal specification language. Model checking is employed for the automatic verification of Formal Tropos specifications.

Enhanced model [8] provides a conceptual agent based modeling approach to drive the requirements gathering and user-oriented requirement analysis process.

RE from Craft to Discipline [9] provides a synergistic blend of incremental model analysis, obstacle analysis, divergence patterns, heuristic rules, conflict links, threat model synthesizer and refinement trees for analysis, conflicts resolution, traceability and documentation of requirements.

NorMAS-RE [10] offers a structured approach to requirements analysis, using graphical notation based on dependence networks and is illustrated in the scenario of virtual organizations based on a Grid network.

B-Tropos [12] framework is delineated as the propagated form of Tropos. In this methodology, agent-oriented requirements have been augmented by business constraints using ConDec graphical language. This framework models various kinds of inter-agent dependencies using abductive logic programming based SCIFF specifications.

RE4Gaia [11] is a requirements modeling approach that extends the Gaia methodology and focuses on dealing with the organizational structure as a means to adequately capture and understand required roles and associated functions in the context of MAS.

Iterative REF [13] supports most importantly four main phases of RE i.e. elaboration, verification, modification and validation using formal concept analysis (FCA) characterized by FCA lattice and CK theory. The normalized class models, use cases, conceptual models and expert domain knowledge are employed as the key concepts in the prescribed approach.

An evaluation of all the prescribed RE frame works including the proposed approach was carried out. A comparative analysis of these RE approaches is listed year wise in table 1. It was observed that a number of prescribed RE approaches are the rich sources of RE techniques facilitating the various phases of RE.

The RE approach i* is well known being one of the most cited methodologies, but facilitates only the very early phase of RE, while Tropos [16] is suitable for the early as well as late phases of requirements analysis of Agent-Oriented Systems. Formal Tropos augments the formal specification and verification using temporal specification language and dynamic modeling. B-Tropos methodology propagates the traditional Tropos methodology by augmenting process-oriented and goal directed temporal business constraints.

The RE approaches viz. GAIA, MaSE, AGORA, RE for large scale MAS, Enhanced model, NorMAS-RE, RE4Gaia and Iterative REF most importantly describe the requirements analysis phase of RE. Requirements traceability is prescribed in few approaches viz. RE: From Craft to Discipline, RE4Gaia, MaSE. Few of the frameworks exhibit the social behavior by modeling inter-agent dependencies using various notations, diagrams and specification languages.

Hence it is observed that a number of RE approaches have been in literature having their salient characteristics. However, one RE method is not always perfect to meet the demands of all software projects. Identification of an appropriate RE method always depends on the nature of the application, size of the project and many other attributes.

5. CONCLUSIONS

There exist a number of approaches for RE. However, all are not suitable for every kind of application; therefore this work investigates various RE approaches taken from literature and illustrates their comparative analysis which would further assist developer in identifying most appropriate method for an application.

6. REFERENCES

- [1] Yu, Towards Modelling and Reasoning Support for Early-Phase RE, IEEE, pp. 226-235, 1997.
- [2] Michael Wooldridge, The Gaia Methodology for Agent-Oriented Analysis and Design, vol. 3, 285-312, 2000
- [3] Mark F. Wood, An Overview of the MAS Engineering Methodology, Springer Verlag, Berlin, 2001
- [4] Haruhiko Kaiya, AGORA: Attributed Goal-Oriented Requirements Analysis Method, IEEE, 2002, PP. 13-22
- [5] Luiz, RE for Large-Scale MAS, Springer, PP. 39-56, 2003
- [6] P. Bresciani, "Tropos: An Agent-Oriented Software Development Methodology", Autonomous Agents and MAS, 8(3). 2004
- [7] Ariel Fuxman; Specifying and analyzing early requirements in Tropos, Springer, RE Journal, 2004
- [8] Prabhat, An Enhanced Model For Agent Based Requirement Gathering And Pre-System Analysis, IEEE, 2006, PP. 187-195
- [9] Axel van Lamsweerde, Requirements Engineering: From Craft to Discipline, 2008, ACM-SIGSOFT, PP. 238-249,
- [10] Serena, NorMAS-RE, springer 2009
- [11] David, A RE Approach for the Development of MAS, International Journal of SE and Its Applications, 2010
- [12] Marco, Engineering and verifying AOR augmented by business constraints with B-Tropos, Springer, pp. 193-223, 2011
- [13] Jonas Poelmans, An iterative REF based on Formal Concept Analysis and CK theory, Elsevier, 2012, 39 (2012) 81158135
- [14] Vibha Gaur, Anuja Soni and Punam Bedi, An Agent-Oriented Approach to RE, 2010, PP. 449-454, USA: IEEE
- [15] Vibha Gaur, Anuja Soni, An Integrated Approach to Prioritize Requirement Using Fuzzy Decision Making, IACSIT, Vol.2, No.4, 2010 ISSN: 1793-8236, pp. 320-328
- [16] Vibha Gaur, Anuja Soni, A Novel Approach to Explore Inter-Agent Dependencies from User Requirements, Elsevier, The journal of Procedia Technology, Nov. 23-27, 2011
- [17] Vibha Gaur, Anuja Soni, Analytical Inference Model for Prediction and Customization of Inter-Agent Dependency Requirements, ACM SIGSOFT Software Engineering Notes
- [18] Vibha Gaur, Anuja Soni, A Fuzzy Traceability Vector Model for Requirements Validation, IJCAT, Inderscience, Special Issue on: "Advanced Software Engineering and Its Applications", 2012
- [19] Ian Sommerville, Software Engineering, Pearson Education, ISBN: 978-81-7758-530-8
- [20] Nuseibeh, B., Easterbrook, S., Requirements Engineering: A Roadmap, Proceedings of the Conference on the Future of Software Engineering, ACM, pp. 35-46.
- [21] Cysneiros, L., M., Yu, E., Requirements Engineering for Large-Scale Multi-Agent Systems, Book: Software Engineering for Large-Scale Multi-Agent Systems, Springer, ACM digital library, pp. 39-56, 2003.
- [22] Jalote, P., An Integrated Approach to Software Engineering, Third Edition, Series: Texts in Computer Science, Springer Verlag, Number of pages 566, 2005.
- [23] Pandey, D., Suman, U., An Effective Requirement Engineering Process Model for Software Development and Requirements Management, 2010 International Conference on Advances in Recent Technologies in Communication and Computing, IEEE, ACM digital library, 2010.
- [24] Nguyen, T., H., Vo, B., Q., Lumpe, M., Grundy, J., KBRE: A Framework for Knowledge-Based Requirements Engineering, Software Quality Journal, Springer, DOI 10.1007/s11219-013-9202-6, 2013.

- [25] Kaindl, H., Is object-oriented RE of interest, Journal of Requirements Engineering, Springer, Vol. 10, Issue 1, pp. 81-84, 2005.
- [26] Rehman, Ur., N., Bibi, S., Ashgar, S., Fong, S., Comparative study of Goal-Oriented Requirements Engineering, 4th International Conference on New Trends in Information Science and Service Science (NISS), IEEE, 2010, pp. 248-253.
- [27] Rashid, A., Aspect-Oriented Requirements Engineering: An Introduction, RE '08. 16th IEEE International Requirements Engineering, pp. 8-12, 2008.
- [28] Rashid, A., Aspect-Oriented Requirements Engineering: An Introduction, RE '08. 16th IEEE International Requirements Engineering, pp. 8-12, 2008.
- [29] Blanes, D., Insfran, E., Abrahao, S., A Requirements Engineering Approach for the Development of Multi-Agent Systems, International Journal of Software Engineering and Its Applications, Vol. 4, No. 2, pp. 1-14, 2010.

Table 1. A Comparative Analysis of RE Approaches

Sr. No.	RE approaches	Year wise publications	Concepts	Support for		RE life cycle coverage	Technique
				Tool for empirical study	Example		
1.	i*	1997, IEEE	SD & SR models, inter-actor dependencies viz. goal, task, soft goal & resource	N.A.	Computer based meeting scheduler	Early phase of RE (Refinement of initial requirements)	Graphical representation
2.	GAIA	2000, Kluwer Academy	Macro-level (societal) and the micro-level (agent) aspects-roles, permissions, responsibilities, protocols, activities, services, acquaintances	N.A.	Agent-based business process management system	Requirements analysis	Role/Organization based approach
3.	MaSE	2001, Springer	Agents, roles, goals, concurrent tasks	CASE-Tool agent-Tool	Open & distributed agents' communication environment	Requirements analysis (capturing goals, applying use cases & refining roles), traceability	Graphical representation using tree like structure, use cases, sequence diagrams & role models
4.	AGORA	2002, IEEE	Refinement of goals by analyzing the impact of changes	N.A.	User account system	Requirements analysis, requirements specification	Preference matrix, contribution value, AGORA graph, quality metrics
5.	RE for large scale MAS	2003, Springer	Lexicon control, strategic actor, first cut model of the domain, autonomy, intentionality and sociality	Language Extended Lexicon (LEL) to understand the vocabulary	Health care domain	Requirements elicitation, analysis & modeling	Health coverage & depth-wise coverage approach, structured analysis & design technique
6.	TROPOS	2004, Kluwer academy	Actor, goal, dependency & plan modeling	JACK-Tool	eCulture system	Early & late phase RE (goal analysis, assigning functions & services	Actor diagrams & goal diagrams

7.	Formal TROPOS	2004, Springer	Actor, goal, strategic dependency, dynamic modeling	T-Tool	Course-exam management	Specification, verification	i* concepts with temporal specification language
8.	Enhanced model for requirements gathering & pre-system analysis	2006, IEEE	Production model, organization model, system user model & cognitive model	N.A.	Agent-based open and adaptive system	Requirements gathering & analysis	Conceptual agent-based modeling
9.	RE: From Craft to Discipline	2008, ACM	Agents, objects, operations, behaviors, multi-view modeling, refinement checking, checking operationalization, risk analysis, threat analysis, conflict analysis, goal-oriented animation, reasoning about alternative options	FAUST (formal analysis suite), animator, SAT-solver Tool	Numerous examples including train control system, phone system on T.V. cable etc.	Analysis, conflicts resolution, traceability and documentation of requirements	Incremental model analysis, obstacle analysis, divergence pattern, heuristic rules, conflict links, threat model, synthesizer, refinement trees
10.	NorMAS-RE	2009, Springer	Agent view, institutional view, role assignment view, combined view, dynamic & conditional dependency modeling, institution, obligation, sanction & secondary obligation	N.A.	The scenario of virtual organizations based on grid network	Requirements analysis	Graphical notation using dependency networks
11.	B-TROPOS	2010, Springer	Concepts of TROPOS augmented with process-oriented & goal-directed temporal business constraints	SOCS-SI, a JAVA based tool is suggested	Product development process scenario	Modeling, specification & verification of requirements	CoDec-graphical language, SCIFF-specifications
12.	RE4Gaia	2010, International Journal of Software Engineering and Its Applications	Mission statement, functional refinement tree, requirements role model, domain model, activity diagram, environmental model, organizational rules	N.A.	Auction System for a Fish Market	analysis, traceability	Graphical representation based on organizational modeling approach
13.	Iterative REF	2012, Elsevier	normalized class models, use cases, conceptual models,	N.A.	book Trader System, hotel administration	elaboration, verification, modification	formal concept analysis (FCA) using FCA

			expert domain knowledge		system, ordering system for computer hardware, office material, etc. of the university KULeuven in Belgium, elevator repair system	& validation	lattice& CK theory
--	--	--	----------------------------	--	---	-----------------	--------------------