

Guidelines From Organizational Requirements to Formal Specification

Fernanda M. R. de Alencar

UFPE - Universidade Federal de Pernambuco

CT - Departamento de Eletrônica e Sistemas
=

Rua Acadêmico Hélio Ramos, s/n - Cidade Universitária

Recife - PE - CEP: 50.740-530

E-mail: fmra@di.ufpe.br

Jaelson F. B. Castro¹

UFPE - Universidade Federal de Pernambuco

CCEN - Departamento de Informática

Av. Prof. Luiz Freire, s/n - Cidade Universitária

Recife - PE - CEP: 50.740-540

E-mail: jbc@di.ufpe.br

KEY WORDS: Organizational Learning, Requirements Formalization, Formal Specification

ABSTRACT

In this work we present some guidelines for the integration of organizational requirements and functional requirements of system. For the organization modeling we use the i* technique, it allows a better description of the organizational relationships among the various agents of a system as well as an understanding of the rationale of the decisions taken. For the formal functional specification of the requirements we use at present Structured Modal Action Logic (MAL). We demonstrate the approach by means of an example of a mineral water factory.

1. **INTRODUCTION**

Requirements Engineering (RE) is the crucial phase in the life cycle of the development of software system. It deals not

only with technical knowledge but also with organizational, managerial, economical and social issues. The Requirements Engineering goals include: (i) the proposal of communication techniques to facilitate information acquisition; (ii) the development of techniques and tools for adequate and precise requirements specification; (iii) to consider alternative requirements specification; and (iv) the possibility of deriving executable specifications that would allow the rapid production of prototype system. As a result, in the recent years we have experienced the proposal of various techniques (some with a rich ontology and diversity of constructors) with a great power of expression and formality. They are capable of improving the precision of the description of stakeholders specification and as a result have greatly improved the quality of requirements specification. However, the tasks of requirements capture and modeling are not easy, because in general, stakeholders do not know, precisely what they want [1]. The majority of existing requirements techniques do not deal with initial requirements that helps to model and analyze the intentions and wishes of the stakeholders. The more complex the problem domain, the more evident the need for techniques capable of representing knowledge and supporting the reasons and

¹ Supported by CNPq grants n. 301052/91-3 (RN) and 520674/93-6 (RN)

motivations involved. Only recently goal oriented techniques that support multiple agents have been proposed in the literature. Among these, we choose the i^* approach [2].

In our work we address the issue of requirements formalization. In particular we show some guidelines on how to transform organizational requirements (i^*) into a pre-formal specification (Structured Modal Action Logic [3] - MAL++) to provide an initial systems specification. Throughout the paper we make use of a mineral water factory as an example, to describe the approach and benefits.

In Section 2, we provide a brief revision of the main characteristics of Structured Modal Action Logic. Section 3 presents the main characteristics of i^* framework. Section 4 presents the organizational model of the case study. In Section 5 we give some guidelines to integrate the organizational approach with the functional one. Section 6 provides some related works. Section 7 concludes the paper with a discussion of the contribution.

1. STRUCTURED MODAL ACTION LOGIC

In this section we review the main concepts of Structured Modal Action Logic - MAL++ [3] and show how it can be used in the description of the behavior of objects (agents) of a mineral water factory.

Modal Action Logic - MAL is based on Typed First Order Logic. It includes types, variables, logical symbols, predicates, functional symbols, constant symbols, terms, atomic formulas and a number of axioms and inference rules. MAL++ is an extension of MAL that has added: (a) pre-defined types called "agents" and "actions" that respectively define real world entities and describe processes that the agents can execute; (b) the modality $[]$ to capture the effect of the occurrence of actions, i. e. $[\text{action } a]$ can be considered as the post-condition or result of an action a that has been completed; (c) deontic operators **per** (permission) and **obl** (obligation) that allow the control over what action can be executed by the agents; (d) combinators; and (e) interval temporal logic of branching linear type. Therefore, a structured MAL specification corresponds to a set of agent (object) descriptions, where the descriptions consists of a set of declarations and axioms that define the behavior of the agents that can interact sharing attributes and actions (label **s** - shared). Some attributes or actions have only local effects and are labeled **l** - local.

1. THE i^* TECHNIQUE

In this section we will review the main concepts of the i^* technique [2]. Usually when we try to understand an organization, the information captured by standard models (DFD, ER, Statechart, etc.) is not enough because the majority of these models describe only entities, functions, data flows, states of system. They are not capable of expressing the reasons and "whys" of the process (motivations, intentions and rationales). The ontology of the i^* technique [2] caters for some of these advanced concepts and can be used for: (i) obtaining a better understanding of the Organizational relationships among the various system agents; (ii) understanding the rationale of the decisions taken; and (iii) illustrating the various characteristics found in the early phases of requirements specification [4]. According

to this technique, the participants of the organizational setting are actors with intentional properties, such as, goals, beliefs, abilities and compromises. These actors depend upon each other in order to fulfill their objectives and have their tasks performed. The i* technique consists of two models: The Strategic Dependency Model (SD) and the Strategic Rationale Model (SR). In the sequel we describe the characteristics of these models, further details can be found in [2] or [4].

The Strategic Dependency Model (SD) consists of a set of nodes and links connecting them, where nodes represent actors and each link indicates a dependency between two actors. Hence, a process is described in terms of network of dependency relationships among various actors, capturing the motivation and why of activities. We can distinguish four types of dependencies, three of them related to existing intentions - **goal dependency**, **resource dependency** and **task dependency** - while the fourth is associated with the notion of non-functional requirements, the so called **soft-goal dependency**. In the **goal dependency**, an agent depends on another one to provide the desired condition, and it does not worry about how this condition is achieved. In the **resource dependency**, the agent depends on the availability of physical resource or information. In the **task dependency**, the agent informs the other what (and how) should be done. The **soft-goal dependency** is similar to the **goal dependency**, except that the condition is not precisely defined at the start of the process, i.e., the goals in a sense involves subjective aspects, that gradually are clarified during the development process. This type of dependency provides an important link connecting two important aspects in software engineering: (i) the technical and (ii) managerial side. We still can identify different degrees of dependencies: open, committed and critical [4]. We can also distinguish **agents**, **roles** and **positions** [6].

The second model is the Strategic Rationale Model (SR) that is used to: (i) describe the interests, concerns and motivations of the process participants; (ii) enable the assessment of the possible alternatives in the definition of the process; and (iii) research in more detail the existing reasons behind the dependencies between the various actors. Nodes and links also compose this model. It includes four types of nodes (also based on the previous SD model): **goal**, **task**, **resource** and **soft-goal**. There are two types of relationship, **means-end** that suggests that there be other means of achieving the objective and **task-decomposition** that describes what should be done in order to perform a certain task.

The i* framework, provides a description, that can be further refined if necessary, of the intentional aspects of the actors that are considered strategic for the clear and complete comprehension of software process.

4. THE CASE STUDY: MINERAL WATER FACTORY

In this section, we present a case study using as example of real mineral water factory. Our intention is to present the i* framework and to migrate from an organizational model to a functional model that is a more precise and complete specification of the system. First we present the Strategic Dependency (SD) model and then we construct the Strategic Rationale (SR) model.

In the Figure 1, we have the SD model of the mineral water factory. The Client wishes to have his bottles filled with mineral water (**goal-dependency**) as well as to have it done safely (**softgoal-dependency**), i.e., that the water to be appropriate to drink. The Company wishes the satisfaction of its client concerning the quality of its services (**softgoal-dependency**). In order to maintain its costs the Company wishes the Client to pay for its services (**resource-dependency**). From the Client's viewpoint he/she wishes to be serviced quickly (**softgoal-dependency**) by the Worker and to receive his/her bottle filled with mineral water (**resource-dependency**). For this the Client depends on the Worker to pass the bottles according to some procedures (**task-dependency**). The agent Worker also depends on the Company to maintain safe his working conditions (**softgoal-dependency**) and at the same time the Company depends on the Worker to maintain the quality of the service (**softgoal-dependency**). The Worker depends on the Company for wages (**resource-dependency**). The Worker depends on the Client to delivery the bottles empty (**resource-dependency**) and in a good state (**softgoal-dependency**).

In our example, the agent Worker has several roles. There are intentional relationships among agents and roles as well as among roles and roles. Due to space limitations we will only describe one agent (see Figure 2). The chosen agent is the worker that has the role of "Do External Cleaning". This agent depends on other agent ("Do Inspection") to pass the dirty and empty bottle that was counted (see Figure 1). We now realize that the initial model needs to be refined to include a new dependency "Counted Bottles" (*resource dependency*) (see Figure 2). The agent "Do External Cleaning" is responsible for receiving the counted bottles, and doing the visual inspection task (a *task-decomposition* of the task "Receive Bottles"). The visual inspection consists of two alternatives: (i) If there is any problem with the bottle this agent will "Return the Bottle" to the Client; (ii) else he will make the cleaning of the bottle (here we can see a *means-end* relationship). The task "Make Cleaning" consists of: removing the residue of the company label ("Remove Label"), sealing wax ("Remove Sealing Wax") and the bottle's cap ("Remove Cap"). It also passes the cleaning bottles ("Pass Cleaning Bottles") to the agent "Do Washing", as can be seen Figure 2. Again we have to add a new resource in the our original model, i. e. "Cleaning Bottles" (*resource dependency*).

Many other details are also important, but due space limitation we will fix our attention in the agent "Do External Cleaning" making the integration with MAL in the next section.

1. GUIDELINES ON THE INTEGRATING I* AND STRUCTURED MAL

In this section, we deal with the question of formalization of requirements expressed in i* technique. We show some heuristics of how we can integrate the i* technique (that express organizational and non-functional requirements) with the structured MAL (used for specifying functional requirements of the system). With both techniques we increase our confidence in the MAL specification, establishing relationships between fragments of the formal specification with some organizational goal described in the i* models. Although the process of obtaining functional requirements from organizational goals is not that obvious, we try to establish some guidelines to help us in the integration process.

Part of the concepts employed in the i* technique can be directly translated into structured MAL

□

Fig. 1 - Strategic Dependency Model of the Mineral water factory

□

Fig. 2 - Strategic Rationale Model of the Mineral Water Factory

Guideline G1:

Agents in the strategic models can be translated as **Objects** in MAL. For example, Agent "DoExternal Cleaning" corresponds to Object "DoExternal Cleaning".

□
=

Figure 3 - Object Definition

Guideline G2:

The *resources* will be mapped :

=

G2a: into *attributes* of MAL **objects** ; and/or

For example, resource "Bottle Empty" correspond to a ttribute "BottleEmpty".

G2b: into **parameter** of the object=92s actions= =2E

For example, parameter of action "ReceiveBottles" is= "BottleEmpty".

□

Figure 4 - Object=92s Attributes and Actions

Guideline G3:

The **tasks** will be mapped into *actions* in MAL. That **axioms** will describe their effect. We can = observe in Figure 2, Agent "DoExternalCleaning", that the task "ReceiveBo= tles" requires a "VisualInspection" that might result in "MakeCleaning" = or "ReturnTheBottles". These tasks will be mapped (see Figure 5) into the= corresponding **actions**. Also note in Figure 2 that the task "MakeC= leanig" can be decomposed into four sub-tasks (RemoveLabel; RemoveSealing= Wax; RemoveCap; and PassCleaningBottles). Due to space limitation we have= not translated them in Figure 5, neither dealt with exceptions and other= more complex situations. For other case studies see [7] [8].

In the real world, during the requirements eng= ineering process the engineer can iterate navigating between the i* model= s (SD and SR) and the MAL specification, enabling him/her to cope with th= e impacts of each description. The two levels of description (organizatio= nal and formal specification) have enabled us to adopt different concepts= of agent at each level, each one is appropriate to the kind of modeling = and reasoning required at each stage. At the level of organizational rela= tionships, it is necessary a notion of agent that is flexible enough to e= xpress the freedom that agents have to violate restrictions and commitmen= ts. At the level of formal requirements specification, a prescriptive vie= w is more appropriate than a descriptive one.

1. RELATED WORK

Yu has shown that cooperating agents [2] can be used= to provide a good understanding of objectives and organizational relatio= nships as well as for the specification and analysis of requirements spec= ifications.

□

Figure5 - Object=92s Axioms

Another related work is that of Yu, Du Bois, D= ubois and Mylopoulos [4], in which they relate the i* to the ALBERT [9] f= ramework. The authors describe how the Albert and i* can work together. T= hey adopt explicitly a set of intentional concepts with a precise semanti= cs related to the ALBERT language.

In KAOS [10] all goals are explicitly modeled and ar= e simplified (reduced) through means-end reasoning until it reaches the a= gent level of responsibilities. Agents should behave as prescribed, which= makes it difficult to analyze strategic relationships and implications. <= /P>

Various other organizational modeling techniques and= formalization have been proposed in the literature [5] [11] [12]. The de= pendency concept has also been used for the coordination of organization = [13].

One of the reasons for choosing the MAL language in this work is our previous experience with the formalism in the context of the FORMLAB project [14] and MULTIVIEW environment [15] [16].

1. FINAL CONSIDERATIONS

The need for modeling the environment is well recognized in Requirement Engineering. Enterprise and organizational models have long been developed. The need for better precision, completeness and consistency of requirements has led the proposal of many tools and techniques. However, when developing system that truly fulfill the real needs of an organization it is required to have a deeper knowledge of intentional and strategic aspects of the agents of the system. Many requirements models can not cope with the questioning of the reasons (or why) and end up dealing only with the functions of the system. The understanding of the rationale related to the agents of the system are important, not only to help, in first instance, in the development of a successful system, but also to facilitate the development of cooperation among the system as well as in the evolution of the system under development.

The *i** technique has provided some means for modeling adequately the requirements of a system, dealing mainly with non-functional aspects that traditionally have not been well represented in the existing conventional models. However, for functional requirements description, another technique is required. In this work we gave some initial guidelines to integrate the descriptive model of the *i** technique with the prescriptive model of the structured MAL logic, which enable a more precise specification in terms of an action logic. The benefits are many and include the possibility of the animation of specification that would help the validation of requirements, and the potential for formal reasoning of the desired system properties.

Some tools support these techniques. For the *i** technique there exists the OME, while for the MAL language, the MULTIVIEW environment [16] allows the generation of MAL specifications, guided by the VSCS Method. However, future work is still required for the integration of these tools under a single environment.

8. REFERENCES

[1] Berry, Daniel. "Myths and Realities in Software Engineering". Tutorial Notes, X Brazilian Software Engineering Symposium, Fortaleza, CE, Brazil, 50 pages, Oct., 1997.

[2] Eric S. K. Yu, "Why Agent-Oriented Requirements Engineering". REFSQ-9297 - Requirements Engineering: Foundation of Software Quality, Barcelona, June 1997.

[3] S. Kent, T. Maibaum, and W. Quirk. "Formally Specifying Temporal Constraints and Error Recovery". Proceedings of IEEE International Symposium on Requirements Engineering - RE93, p.208-215, Jan. = 1993.

[4] Eric Yu, Philippe Du Bois, Eric Dubois, John Mylopoulos, "From Organization Models to System Requirements - A Cooperating Agents Approach". The 3rd International Conference on Cooperative Information Systems - CoopIS-95, Vienna (Austria), May, 2-9, 1995.

[5] Janis A Bubenko. "On Concepts and Strategies for Requirements and Information Analysis". In Information Modeling, p. 125-169. Chartwell-Brant, 1993.

[6] Eric S. Yu. "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering". Third

IEEE International Symposium on Requirements Engineering - RE-9297, Annapolis, USA, Jan. 1997, pp. 2E

[7] F. M. R. Alencar and J. F. B. Castro. "Utilizando a Lógica Modal de Ações para Formalização de Requisitos Organizacionais (in Portuguese)". In Proceedings of XXI Conferência Latinoamericana de Informática, Valparaíso, Chile, Nov. 1997, pp. 515-524.

[8] F. M. R. Alencar and J. F. B. Castro. "Formalização de Requisitos Organizacionais (in Portuguese)". In Proceedings of Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software, Torres, RS, Brazil, April 1998.

[9] E. Dubois, P. Du Bois and M. Petit. "Object-Oriented Requirements Analysis: an Agent Perspective". In O Nierstraz, editor, Proc. Of the 7th European Conference on Object-Oriented Programming - ECOOP-9293, pp. 458-481, Kaiserslautern, Germany, Jul. 26-30, 1993.

[10] A Dardenne, A van Lamsweerde, and S. Fickas. "Goal-directed Requirements Acquisition". Science of Computer Programming, 20:3-50, 1993.

[11] Roel Wieringa and E. Dubois. "Integrating Semi-Formal and Formal Software Specification Techniques". In Matthias Jarke (Germany) and Dennis Shasha (USA) editors, Information Systems Vol. 23, No. 2/4, pp. 159-178, May/June 1998, Published by Elsevier Science Ltd.

[12] Emanuele Ciapessoni, A Coen-Porisini, E. Crivelli, D. Mandrioli, P. Mirandola, A Morzenti. "From Formal Models to Formality-based Methods: an Industrial Experience". To appear in ACM Transactions on Software Engineering and Methodologies.

[13] Thomas W. Malone and Kevin Crowston. "The Interdisciplinary Study of Coordination". Computing Surveys, 26:87-119, Mar. 1994.

[14] J. F. B. Castro. "The Process of Requirements Formalization: The FORMLAB Project". In Proceedings of Information Systems Analysis and Synthesis - ISAS-9295, 5th International Symposium on Systems Research, Informatics and Cybernetics, August, 16-20, 1995, Baden-Baden, Germany, pp. 01-05.

[15] J. F. B. Castro, C. J. Gautreau and M. A Toranzo. "Tool Support for Requirements Formalization". In Proceedings of the ACM SIGSOFT Viewpoints 96: International Workshop on Multiple Perspectives in Software Development, San Francisco, USA, October 1996, pp. 202-206.

[16] J. F. B. Castro and M. A Toranzo. "Towards Software Quality: The Multiview Case". REFSQ-9297 - Requirements Engineering: Foundation of Software Quality, Barcelona, June 1997.