

A Novel Organizational Model for Real Time MAS: Towards a Formal Specification

Mohamed Amin Laouadi¹, Farid Mokhati², and Hassina Seridi³

¹ Computer Science Department, Ferhat Abbas University, Setif -1-, Algeria
Laouadiamin@yahoo.fr

² Computer Science Department, Larbi Ben M'hidi University, Oum El Bouaghi,
Algeria LAMIS Laboratory, Tebessa University
mokhati@yahoo.fr

³ Computer Science Department, Badji Mokhtar-Annaba University,
P.O. Box 12, 23000 Annaba, Algeria LabGED Laboratory
seridi@labged.net

Abstract. In this paper we present our approach allowing the translation of Real Time Multi-Agents Systems (RT-MAS) organizational requirements described by extended AUML (Agent UML Language) diagrams into a formal specification written in Real Time Maude language (RT-Maude). In fact, the approach is an extension of our previous work [1] that consists in extending AUML diagrams (Temporal AUML organization use case diagram and Temporal AUML organization class diagram) by using stereotypes notions and meta-model organizations entities for taking into account RT-MAS specificities. Once elaborated, these different diagrams undergo a validation to assure inter-and intra model coherence. The formal and object oriented language RT-Maude, base on rewriting logic, supports formal specification and programming of concurrent systems. The main motivations of this work are: (1) formalizing the organizational requirements of RT-MAS by using RT-Maude language, and (2) integrating the validation of the coherence models, since the analysis phase.

Keywords: Organizational Requirements; Formal Framework; Agent UML; Rewriting Logic; RT-Maude.

1 Introduction

Currently, Agent Oriented Software Engineering (AOSE) is a very active research domain. For many years, MAS designers have development methodologies and modeling language without reflects the different temporal restrictions that RT-MAS may have. Indeed, it is not easy to conceptualize RT-MAS concepts using conventional agent software engineering approaches. Consequently, a critical research issue for the real time agent community has been the definition of suitable organizational model for analyzing and designing their main properties.

Moreover, even the proposed methodologies for the development of MAS as stressed in our previous work [1] and those proposed for RT-MAS development: (like

'RT-Message' [2], 'BDI-ASDP extended for real time' [3] and 'Development Method of Lichen Zhang' [4]), are inadequate. They have certainly made important responses in the development process of RT-MAS. However, the methodological aspect is not yet mastered. Indeed, none of these methodologies take into account the formalization of the organizational requirements for the future system. The quality of model analysis has an extreme importance for the remainder of the development process phases. Their formal specification and validation allow avoiding many problems that may affect the development quality.

Formalizing the organizational aspects of RT-MAS is in our opinion, an importance way for both analysis and design activities. Furthermore, the RT-MAS design requires the involvement with formal languages. Among these languages: RT-Maude [5] is probably the best known and most widely used languages for object oriented formal specification. There is currently no work applying RT-Maude to RT-MAS formal organizational specification and both to real time applications.

The present work takes place in the context of our project, whose objective is to develop a generic organizational formal framework for organization- oriented specification of RT-MAS aspects.

As first step in our project, we have presented in a precedent work [1], extensions made on AUML diagrams for describing Real-time Multi-Agent Systems organizational requirements.

However, AUML models [6] suffer of a lack of formal semantics. It may contain inconsistencies which are difficult to detect manually. Formal methods represent an interesting solution to this problem. The formal specifications will have the effect of eliminating the ambiguities in the models interpretation. For that, the combination of AUML and RT-Maude will formally validate the developed organizational model.

This work presents a systematic approach supporting the translation of RT- MAS organizational requirements represented by extending AUML diagrams [1] into a formal specification writing in RT-Maude language. This last is multi-paradigms language which combines the functional programming and object-oriented programming. Furthermore, RT-Maude is very powerful in terms of specification, validation and verification of concurrent systems, making it a good candidate for specification and validation of RT-MAS.

The aim of this approach is to translating extended AUML diagrams into a RT-Maude formal framework to integrate the formal validation of the consistency of the organizational model, since the analysis phase.

The remainder of this paper is organized as follows. Section 2 presents a general overview of similar works. In section 3 we give a survey of used diagrams. In section 4 we give a brief overview of rewriting logic and RT-Maude language. The proposed approach and the translation process are presented in Sections 5. In section 6 a discussion of our contribution is given. Finally, we give a conclusion and some future work directions in section 7.

2 Related Works

Several methods to modeling organization oriented MAS have been analyzed [1], such as AGR [7], Moise+ [8], INGENIAS [9], and Gaia [10]. They have modified the MAS construction through giving rise to new organizational concepts, but without taking into account the agent temporal behavior. For this reason we will present briefly in this section three methodologies that directly addressing the design of real-time multi-agent systems, we are interested by: the Methodology RT-Message [2], the extended BDI-ASDP methodology for real time [3] and the development method of Lichen Zhang [4]. For a description of real-time agents, these three methodologies use different models namely: domain model, role model, and timed model (Table 1.).

Table 1. Real-Time Agent Models Identification Approaches.

	Domain Model	Role Model	Timed Model	Organizational Model
The RT- Message Methodology [2]	✓	✓	✓	very simple model
Extended BDI ASDP methodology for real time [3]	✓		✓	
Zhang development method[4]	✓		✓	

In RT-MESSAGE, an organization model is built viewing the system as a single entity. The model shows its interactions within the environment, identifying the events which it must react to. It is also necessary to estimate approximate response times for all the identified events. In our opinion, this model is abstract and still remains largely unresolved, which we believe should be described more clearly and deeply. Also, the concept of role is only present in RT-Message methodology [2], where the roles are identified independently of the agent system.

Regarding the modeling of temporal constraints of real-time agents, each methodology offers an approach: for the RT-Message case, extensions made on the different models imported from the MESSAGE method [11] [12] allow analyzing the MAS for real-time environments. For example, "the Goal / Task model" has been modified to incorporate a taxonomy of goals (Goal taxonomy) which takes into account temporal criteria. When specifying the goal's different types, it is necessary to extend the goal and task patterns of the method "message" for integrating the real-time features. The artifacts obtained are a set of 'implications diagrams' showing the relationship between goals and tasks. Subsequent, in extended BDI-ASDP for real time, proposed by Melián et al. [3], the temporal constraints modeling is done through "the timing diagrams" specified in UML 2.0. To satisfy the need to model real-time systems by the agent approach, Zhang [4] proposed to extend UML by introducing a new stereotype, called <<agents>>. The timing characteristics are specified as an instance of this stereotype called <<TimeAspect>>.

However, the RT-MAS modeling is frequently linked to organizational specifications in the sense that those specifications provide a basis for describing the organizational requirements of agents, applying a set of software engineering techniques. This concept has been neglected in these analyzed methodologies, no one of them takes in to account the temporal dimension jointly with organizational concepts (Table 1).

Although these methodologies brought much important answers in the development process and in particular for describing RT-MAS requirements, they offer only informal or semi-formal descriptions for representing RT-MAS organizational requirements. Our approach offers a joint representation of the organizational requirements while profiting from the advantages of the semi-formal and the formal approaches. Furthermore, the proposed formal approach, allows reducing confusion and misunderstanding risks between developers and users.

3 Used Diagrams

3.1 Temporal AUML Organization Use Case Diagrams

In [1], we have adopted some extensions to conventional use case diagrams (Agent-O Use Case; Temporal Agent-O Use Case; Agent-O; External Agent-O; and Real Time Agent-O) by using the extension mechanism of "stereotyping", to describe roles responsibilities, functions and behaviors in an organization.

3.2 Temporal AUML Organization Class Diagrams

This type of diagrams allows us to reflect the static relationships among roles with taking into account temporal restriction of real time agent in organizations. Our idea was to added new compartments (Temporal Organization; Groups; Social norms; Goal; and Task) to the classical AUML class diagrams based on stereotype concept [1].

3.3 Modeling Dynamic Behaviors of Agent Organizations

The proposed model, describes the desired behavior of the agent organization and its general structure (roles, interactions, and temporal constraints) by means of AUML state-chart and protocol diagrams that describes agents' individual and collective behaviors respectively.

In what follows we briefly present Real-Time Maude language and the basic concepts related to the proposed translation process.

4 Real Time Maude

Maude is a language for specifying and programming systems. It is based on rewriting logic [13] [14] [15]. Maude allows describing easily the intra and inter-object concurrency. In rewriting logic, the logic formulas are called rewriting rules. They have the

following forms: $R: [t] \Rightarrow [t']$ or $R: [t] \Rightarrow [t']$ if C . Rule R indicates that term t becomes (is transformed into) t' . On its second form, a rule could not be executed except that a certain condition C is verified. Term t represents a partial state of a global state S of the described system. The modification of the global state S of the system to another state S' is realized by the parallel rewriting of one or more terms that express the partial states. The distributed state of a concurrent system is represented as a term whose sub-terms represent the different components of the concurrent state [16].

Real-Time Maude [5] is an extension of Maude that was designed to exploit the concepts of the real-time rewrite theory. A real-time rewrite theory is a Maude rewrite theory, which also contains the specification of:

sort Time to describe the time domain,

sort GlobalSystem with a constructor '{_}': $\{ _ \} : \text{System} \rightarrow \text{GlobalSystem}$

And a set of tick rules that model the elapsed time in the system that have the following form: $\{t\} \Rightarrow \{t'\}$ in time if condition μ

Where μ is a term which may contain variables, of sort Time that denotes the length of the rule, and the terms t and t' are terms of sort System, which denotes the state of the system. The rewriting rules that are not tick rules are rules supposed to take a time instant zero. The initial state must always have the form $\{t''\}$, where t'' is a term of sort System, so that the form of tick rules ensures that time flows uniformly in all parts of the system. Real-time rewrite theories are specified in Maude as timed modules or timed object-oriented modules.

5 Translation Process

In this section, we present our approach that allows obtaining a RT-Maude formal specification.

The proposed translation process aims to translate the extended AUML diagrams described above (Section 3) to describing RT-MAS organizational requirements to RT-Maude formal specifications. This process is divided into three major steps (Fig. 1): (1) description of RT-MAS organizational requirements using AUML diagrams, (2) inter-diagrams validation, and (3) generation of RT-Maude formal specification.

The first step is the usual analysis phase of software development process. The second step aims to validate the coherence between the designed models. The last step is the systematic generation of RT-Maude source code from the considered AUML diagrams.

The formal framework proposed (Fig. 2) is composed of several RT-Maude modules: functional modules, object-oriented modules, and timed object-oriented modules. For reason of limitation of space, we present only the main modules of the proposed formal framework. We use the suffix "O" to indicate "organization".

The Module GOAL describes agents' organization goals; the TASK module describes the tasks that an agent-O can perform. These two last modules and the GROUP module (which is used to define the Agent-O groups) are imported into ORGANIZATION module to define the organizations related to agent's goals.

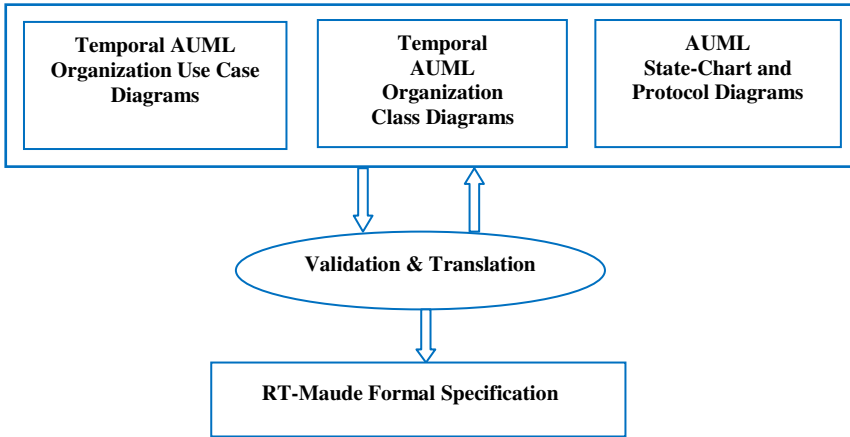


Fig. 1. Overview of the Translation Process

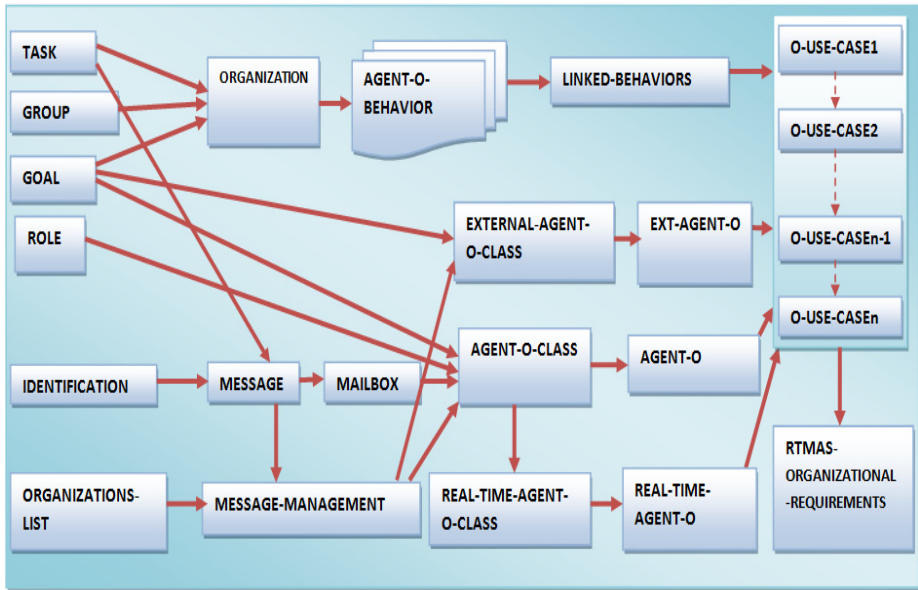


Fig. 2. Generated Modules

AGENTi-O-BEHAVIOR modules that import ORGANIZATION module are used to illustrate the behavior of individual Agents-O. In order to respect interactions between the different Agents-O, connections between them are performed through the LINKED-BEHAVIOR module, which reuses AGENTi-O-BEHAVIOR modules. The identification mechanism for Agents-O is defined by the IDENTIFICATION module, and message structure description exchanged between the various Agents-O is done via MESSAGE module that imports the IDENTIFICATION, and TASK modules.

Communicating agents are generally endowed with a Mailbox containing the received messages of other Agents-O and a list of its organizations. For that, we define the functional modules MAILBOX and ORGANIZATIONS-LIST to manage respectively Mailboxes and organizations lists of agents. Agents-O roles are defined in the module ROLE. To describe the sending/ receiving operations, we define module MESSAGE-MANAGEMENT which imports ORGANIZATIONS-LIST and MESSAGE modules.

The object oriented module EXTERNAL-AGENT-O-CLASS (Fig. 3) is used to define the base class of external agents organization, with attributes *CurrentGoal* and *OrgList* (line [1]) which represent the agent's current goal and its organizations list. This module imports GOAL, and MESSAGE-MANAGEMENT modules.

```
(omod EXTERNAL-AGENT-O-CLASS is
protecting GOAL . protecting MESSAGE-MANAGEMENT .
class ExtAgentO | CurrentGoal : Goal, OrgList :
OrganizationList . ---[1] endom)
```

Fig. 3. The O.O Module EXTERNAL-AGENT-O-CLASS

In the object oriented module AGENT-O-CLASS (Fig. 4), we define the internal agents' organization base class structure. This class (line [1]) has as attributes: *PlayRole*, *CurrentGoal*, *MBox* and *OrgList* to contain in this order: the role played by the agent, its current goal, its mailbox and its organizations list. This module imports all the modules: GOAL, ROLE, MAILBOX, and MESSAGE-MANAGEMENT.

```
(omod AGENT-O-CLASS is protecting GOAL. protecting
ROLE.
protecting MAILBOX . protecting MESSAGE-MANAGEMENT .
class AgentO | CurrentGoal : Goal, PlayRole : Role,
OrgList: OrganizationList, MBox : MailBox.--[1] endom)
```

Fig. 4. The O.O Module AGENT-O-CLASS

To describe the Real-Time Agents Organization, we have defined the *RealTimeAgentO* class with the attribute *Clock* (line [1]) in the timed object oriented module REAL-TIME-AGENT-O-CLASS (Fig. 5) as a subclass of *Agent-O* Class (line [2]).

```
(tomod REAL-TIME-AGENT-O-CLASS is extending AGENT-O-
CLASS .
class RealTimeAgentO | Clock : Time . ---[1]
subclass RealTimeAgentO < AgentO . ---[2] endtom)
```

Fig. 5. The Timed O.O Module REAL-TIME-AGENT-O-CLASS

To each organization use case is associated one timed O.O module O-USE-CASE_i (Fig. 6), which has the same name as the corresponding organization use case. In each module O-USE-CASE_i are defined the rewriting rules describing the different interaction scenarios between the agents-O defined in the different AUML Protocol diagrams,

instances of the organization use case. Note that these rules may be instantaneous rules or tick rules, conditional or unconditional.

```
(tomod O-USE-CASEi is inc EXTERNAL-AGENTS-O . inc
AGENTS-O . including REAL-TIME-AGENTS-O. including
LINKED-BEHAVIORS.
rl [1] : Configuration1 => Configuration2. ...
rl [m] : Configuration 2m-1 =><Configuration2m. endtom)
```

Fig. 6. The Timed O.O Module O-USE-CASEi

Once generated, all O-USE-CASEi modules are imported in the timed object oriented module RTMAS-ORGANIZATIONAL-REQUIREMENTS (Fig. 7) which describes all system's organizational requirements.

```
(tomod RTMAS-ORGANIZATIONAL-REQUIREMENTS is
including O-USE-CASE1. ... including O-USE-CASEm. endtom)
```

Fig. 7. The Timed OO Module RTMAS-ORGANIZATIONAL-REQUIREMENTS

The tick rule used to ensure the progress of time in the system is given in Fig. 8, where we have defined the message *Timer* to change the Real Time Agent Organization clock defined by the attribute *Clock* (line [1]).

```
crl [tick] :{Timer(TimeOut) < A : RealTimeAgentO |
CurrentGoal : G, Clock : T, PlayRole: Initiator> --[1]
REST:Configuration} => { Timer(TimeOut minus 1)
< A : RealTimeAgentO |CurrentGoal: G, Clock : T plus 1>
REST:Configuration } in time 1 if (TimeOut > zero).
```

Fig. 8. The Tick Rule

6 Discussion

As illustrated in our previous work [1], the proposed approach considers jointly functional, static and dynamic aspects of RT-MAS from AUML extended diagrams. Among possible techniques used in the literature to validate and verify informal and/or semi-formal models is that which consists of translating these three views into Real-Time Maude formal descriptions to produce precise descriptions and also offer a better support to their verification and validation processes.

We propose that in the initial requirements phase, an organizational model is defined, detailing the main actors, roles, goals and dependencies. In this model, several organizational elements are considered (such as a simple structure, agent roles, and goals) and during the translation process, we have developed a formal framework, where several modules are generated, like illustrates in Fig. 2, to integrate the formal validation of the consistency of the organizational model, since the analysis phase.

In fact, our idea consists basically in using jointly AUML extended diagrams presented in [1], and RT-Maude formal framework together to support the formal specification of RT-MAS organizational requirements. In this formal organizational framework, the strengths of both approaches are unified, but it still lacks suitable tools for analysis and design. We are currently working on the validation of this generated RT-Maude specifications (modules), from the considered AUML extended diagrams, using a concrete case study.

7 Conclusions

The formalization of organizational requirements represents an important activity during development process of real time multi-agent systems. It produces a rigorous description and offers a solid basis for the verification and the validation activities. Several methodologies describing MAS organizational requirements are proposed.

However, they only offer informal or semi-formal descriptions. In this paper we proposed a generic approach that allows firstly, capturing functional aspect (temporal AUML organization use case diagram), static aspects (temporal AUML organization class diagram), and dynamic aspects (AUML protocol diagrams together with AUML state-chart diagrams) of RT-MAS based on organizational perspective, and secondly, translating the graphical description in a formal description RT-Maude. This later characterizes by the power of description and integrates several tools of verification and validation.

Although, this formal organizational framework represents an important approach for modeling RT-MAS, it is a recent extension which combines the AUML extended diagrams [1], with RT-Maude specification. In this framework, the strengths of both approaches are unified, but it still lacks suitable tools for analysis and design.

As future directions to this work, we plan on the development of additional formal analysis techniques including temporal analysis (using model-checking) to verify some properties of Real-Time Multi-Agent System organizational requirements specification, by integrating possibilities offered by Real-Time Maude.

References

1. Laouadi, M.A., Mokhati, F., Seridi-Bouchelaghem, H.: Towards an Organizational Model for Real Time Multi-Agent System Specification. In: Proc. SAI, October 7-9. IEEE, London (2013)
2. Julián, V., Soler, J., Moncho, M.C., Botti, V.: Real-Time Multi-Agent System Development and Implementation (2004)
3. Melián, S.F., Marsá, I., Ukrania, M., Miguel, D.-R., Carmona, A.-L.: Extending the BDI ASDP methodology for real time (2005)
4. Zhang, L.: Development Method for Multi-Agent Real Time Systems. Faculty of Computer Science and Technology Guangdong University of Technology. International Journal of Information Technology 12(6) (2006)
5. Olveczky, P.C.: Real-Time Maude 2.3 Manual. Department of Informatics, University of Oslo (2007)

6. Odell, J., Parunak, H.V.D., Bauer, B.: Extending UML for Agents. In: Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence. ICue Publishing, Austin (2000)
7. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems, pp. 443–459 (2004)
8. Hubner, J.F., Sichman, J.S., Boissier, O.: A model for the structural, functional, and deontic specification of organizations in multi-agent systems. In: Bittencourt, G., Ramalho, G.L. (eds.) SBIA 2002. LNCS (LNAI), vol. 2507, pp. 118–128. Springer, Heidelberg (2002)
9. Gomez, J., Fuentes, R., Pavon, J.: The INGENIAS Methodology and Tools. Agent oriented Methodologies, pp. 236–276. Idea Publishing Group (2005)
10. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and MAS* 3(3), 285–312 (2000)
11. Message web site: <http://www.eurescom.de/public/projects/P900-series/p907/>
12. Message, Metamodel web site, <http://www.eurescom.de/~public-website/P900-series/P907/MetaModel/index.HTML>
13. Clavel, M., Duran, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.: Maude Manual (version 2.2). SRI International, Menlo Park, CA 94025, USA (2005)
14. Meseguer, J.: Rewriting as a Unified Model of Concurrency. *SIGPLAN OOPS Mess.* 2(2), 86–88 (1991)
15. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Quesada, J.F.: Maude: Specification and Programming in Rewriting Logic. *Theoretical Computer Science* (2001)
16. Mokhati, F., Badri, M., Zerrougui, S.: A Novel Conformance Testing Technique For Agent Interaction Protocols. In: Proceedings of Science and Information Conference (SAI), London (2013)