UML Profile for Modeling Multi Decisional Reactive Agent System

Abdelhay Haqiq and Bouchaib Bounabat

Abstract—A Reactive System is a system engaged in stimulus-response behavior in order to produce desirable effects on its environment at a specific time. The failures of developing such a system are usually caused by design errors despite of several various tests and simulations made by the developer. Indeed, it is essential to avoid these errors at an advanced level of the application's life cycle by using a formal specification and verification method. Moreover, the Multi Decisional Reactive Agent (MDRA) approach is devoted to specify and verify such a system. On the other side, modeling the system processes is primordial to describe either its internal or external behavior; however, MDRA is only supported by a BPMN which models its external activities. For this reason, there is a necessity to make easier modeling the MDRA behaviors by using an appropriate method. Furthermore, UML profile provides an extension mechanism to adapt UML for modeling a particular domain, it also specializes UML on each context and introduces new concepts and rules. In this paper, we propose an approach that combines both formal specification and semi-formal modeling which helps designers to specify their application through an UML profile responding to the characteristics of MDRA.

Index Terms—Reactive, agent, formal specification, UML profile.

I. INTRODUCTION

A Reactive System is a system that reacts continuously to its environment; it is executed with a specific speed determined by that environment. This class of system differs from systems which transform inputs to outputs at their own pace, such as transformational and interactive system [1]. *Transformational system* is a system that starts by executing inputs and, after a certain time, ends by providing outputs. *Interactive System* is a system which is in a continual interaction with its environment but it is running at its own speed. The reactive system can be viewed in different fields [2], such as *real-time system*, which is a system where the validation of a response is relied on the time where the response should be produced. The importance in this case is not limited on just producing outputs but consists on the production of outputs at a specific time; *workflow management system*, the companies such as insurances and banks, use a system that can monitors tasks to be performed in an organization and assign these tasks to employees working in the organization. By doing this, the employees

enforce a certain life cycle on these tasks; *e-commerce system* is an example of an electronic auctioning system, which makes available a distributed infrastructure where consumers can meet one another and negotiating prices, this type of system takes place according to a convinced protocol that imposes a certain behavior on the participants. Thus, due to the complexity of reactive system, such a system requires an appropriate approach in order to limit the introduction of potentially faulty components during the construction of the system. Indeed, the failures of computer applications are usually caused by design errors despite of various tests and simulations made by the developer. It is therefore essential to avoid these errors at an advanced level of the application's life cycle by using a formal specification and verification method. Multi Decisional Reactive Agent (MDRA), is among several methods that contribute to enhance the reactive system behaviors using verification aspect, It is developed by [Bounabat, al.] [3], the MDRA is widely used and applied in different context and domains such as mobile systems [4], [5] and organizational systems [6]. The main component of MDRA is Decisional Reactive Agents (DRA), which can acts in an autonomous way until that their realizations are undertaken in adequate deadlines. On the other side, although the MDRA system can be modeled using BPMN to specify its external behaviors, it doesn't have the mechanisms to model properly its internal activities. For this reason, there is a necessity to facilitate modeling the MDRA behaviors by using an appropriate method. Furthermore, UML profile provides an extension mechanism to adapt UML for modeling a particular domain, it also specializes UML for each context and introduces new concepts and rules.

In this paper, we propose an approach that combines both formal specification and semi-formal modeling which helps designers to specify their application with taking into consideration the aspect of the reactive system, through the specification of the system with Multi Decisional Reactive Agent and modeling such system using UML profile that responds to the characteristics of MDRA.

The remainder of this paper is organized as follows: Section II sets out the formal description of Multi Decisional Reactive Agent System (MDRAS). Section III presents review literatures on using UML in Multi-Agent systems modeling. Section IV exposes the proposal approach MDRA Profile. Section V illustrates the use of the proposed approach by its application to the specification of process acting in case of a humanitarian emergency. Finally, in Section VI, we conclude and propose a global planning about our future work.

Manuscript received January 8, 2013; revised March 20, 2013.

Abdelhay Haqiq and Bouchaib Bounabat are with the ENSIAS (National Higher School for Computer Science and System Analysis), Mohammed V Souissi University (UM5S), Rabat, Morocco (e-mail: abdelhay.haqiq@um5s.net.ma, bounabat@ensias.ma).

II. MULTI DECISIONAL REACTIVE AGENT SYSTEM

Multi Decisional Reactive Agent developed by [Bounabat, al.] is among useful approaches devoted to specify and verify the reactive system modeling. The application of the MDRA is illustrated in [5], [6], where MDRA models and checks the automated systems of production as in the mobile systems domain, and organizational systems. The main component of MDRA is Decisional Reactive Agents (DRA), which can acts in an autonomous way until that their realizations are undertaken in adequate deadlines.

A. Formal Description of a DRA

The formal description of a Decisional Reactive Agent (DRA) is based on 10-uplets noted $\langle A, D, S, E', O, E, O', \text{Act}, \text{Dec}, \text{Sig} \rangle$ (see Fig. 1), where:

- 1) **A**: Set of actions exerted on the agent, each action represents a possible operation to be performed on this agent in order to achieve a specific target
- 2) **D**: Set of decisions generated by the agent so as to bring solutions concerning the system behavior. Each decision d is characterized by its action horizon $H_a(d)$, that indicates the time during which this decision remains valid
- 3) **S**: Set of Signaling received by the agent. The signaling represents the acknowledgement response to confirm the execution of decision
- 4) **E'**: Set of external states delivered by the agent. Each external states represents the object state emitted to the environment
- 5) **E**: Set of agent's internal states. Each one indicates the current state of the agent
- 6) **O**: Set of agent's internal objectives. Each internal objective denotes the expected state after the execution of a decision
- 7) **O'**: Set of agent's external objectives which can be achieved. These objectives represent the agent interpretation of each action, and define therefore its different behaviors

The three last uplets $\langle \text{Act}, \text{Dec}, \text{Sig} \rangle$ correspond to the decisional functions that define the DRA behaviors:

- 1) **Act**: $\forall a \in A, \exists ! o' \in O' / \text{Act}(a) = o' \Rightarrow a \leftrightarrow o'$; means that the occurrence of an action a implies instantaneously the occurrence of its associated external objective o' .
- 2) **Dec**: $\text{Dec}(o', e) = (d, o) \Rightarrow [o' \wedge e \leftrightarrow d \wedge o]$; means that depending on the current external objective o' and as soon as the agent is in an appropriate internal state e , a corresponding decision d and an internal objective o are instantaneously produced by the function **Dec**.
- 3) **Sig**: $\text{Sig}(o', o, s) = (e, e') \Rightarrow [o' \wedge o \wedge s \leftrightarrow e \wedge e']$; means that depending on the current external objective o' and the expected internal objective o , and as soon as the receipt of a signaling s , its associated external state e' is instantaneously emitted and the new agent internal state turns into e .

B. Formal Description of MDRAS

A Multi Decisional Reactive System adopts the hierarchical structure based on two-level tree (see Fig. 2), consisting of DRA Supervisor (DRAS) and two or several

possible sub-agent components (MDRAS_i), the connection between supervisor and its sub-agents is realized through communication interfaces, which is constituted of both Decisional Interface (DI) and Signalization Interface (SI).

The formal description of a MDRAS can be defined as a quadruplet $S : \langle \text{DRAS}, \text{DI}, \text{SI}, \text{S-MDRAS} \rangle$, with:

- 1) **DRAS**: represents the DRA type that supervises S .
- 2) **DI**: Decisional Interface of S , implements a translation function of a decision into several parallel actions, each of these actions is led to an inferior sub-agent level.

Fig. 1. Internal structure of a multi reactive decisional agent.

- 1) **SI**: Signalization Interface of S , implements a translation functions of several external states into one and only signalization.
- 2) **S-MDRAS**: characterizes a set of 2 or more MDRAS components.

Fig. 2. Internal structure of a multi reactive decisional agent.

III. LITERATURE REVIEWS

A. Multi-Agent and UML

Combining both Multi-Agent and UML has been realized in many different projects. The [7] presents an Agent Modeling Language (AML) for specifying, modeling and documenting the social aspects of Multi-Agent System (MAS), the social aspects considered in MAS are social structure, behavior and attitudes, the authors focus on modeling behavior and mental attitudes using UML profile. The [8] developed an UML profile for the modeling of requirements in MAS projects. The profile proposed adapted

the Use Cases diagram for modeling reactive and cognitive agents, as well as actions, perceptions, goals and plans for those agents in collecting and analyzing requirements level. The [9] proposes to use UML 2.1 sequence diagram and composite structure diagram to model social and goal-driven software engineering ontology based multi-agent systems. The [10] presents a UML extension for agent-based system modeling called MAS-ML. The MAS-ML proposes a Class diagram to model the relationships between objects, agents, organizations and environments, while a Sequence diagram models the interactions between agents playing roles, organizations playing roles, environments and objects while either playing roles or not.

Our objective is to simplify the specification of the internal Multi Decisional Reactive Agent System (MDRAS) behaviors using UML profile, which provides an extension mechanism for adapting UML to model particular domain; it specializes UML for each context and introduces new concepts and rules.

B. UML Statechart

One of the most UML diagrams that have the ability to model the MDRA behaviors is the Statechart Diagram since it is used to describe the states of different objects in the system's life cycle; these states are changed when some events are triggered. A Statechart Diagram is characterized by different notions; a *state* is a condition of being at a certain time, a *transition* represents a certain change of states from a source state to a target state, a *trigger* is an event that may cause a transition, a *guDRA* is a Boolean condition that allows or blocks the transition, and an *effect* is an uninterruptible operation which is executed while the transition is taking place [11].

In this paper, we focus on using both trigger and effect explained above, because we assume that the MDRA as it is defined doesn't employ the *guDRA* notion. On the other side, in order to model the different subsystems and define the structure of a system, UML provides the Package Diagram, which gives the ability to the designer to organize models into package.

IV. MULTI DECISIONAL REACTIVE AGENT PROFILE

This section presents the proposal UML Profile for MDRA, based on stereotypes and tagged values. In the table below (Table I), we describe the stereotypes and their notations; also we categorize the specification levels of MDRA Profile.

The first column in the table above indicates the categories' levels for the definition of MDRA Profile, there are four levels. The Multi-DRA structure level defines the interaction between DRA Supervisor and DRA Element, the Internal DRA level specifies the internal behavior of DRA X (X: Supervisor or Element), the External Objective level specifies the internal behavior of External Objective and the Communication Interface level identifies the concurrent and interactive subsystems.

The specification of MDRA Profile begins by modeling the DRA Supervisor which is situated in the top of hierarchical of *Multi Decisional Reactive Agent System*; it is used to master the different Sub-Agent through Action,

Decision and Signaling. The Sub-Agent can be either an **<<DRA Supervisor>>** or an **<<DRA Element>>**, the latter is an agent which the Communication Interface in not included. The relation between DRA is characterized by two aspects, the first one is the notion of **<<use>>** which indicates that DRA Supervisor use that specific DRA X (X: Supervisor or Element). The second notion is **<<reuse>>**; this notion puts in value the aspect of reusing the different components defined in an DRA X. In each DRA X, the tagged value **NumberOfAgent** is mentioned; it indicates the number of occurrences using the same DRA.

TABLE I: MDRA PROFILE DESCRIPTION

	Stereotype	MetaClass	Notation
Multi DRA structure	<<DRA Supervisor>> Tagged value = NumberOfAgent	Package	
	<<DRA Element>> Tagged value = NumberOfAgent	Package	
	<<Use>>	Dependency	
	<<Reuse>>	Dependency	
Internal DRA	<<C_DRA>>	State	
	<<T_Action>>	Transition	
	<<External Objective>>	StateMachine	
	<<T_External State>>	Transition	
External Objective	<<C_Internal State>>	State	
	<<E_Decision>> Tagged value = ActionHorizon	Transition	
	<<Internal Objective>>	StateMachine	
	<<T_Signaling & E_External State>>	Transition	
	<<T_Signaling>>	Transition	
	<<T_AH & E_External State>>	Transition	
	<<T_AH>>	Transition	
	<<Internal State>>	State	
Communication Interface	<<T_Decision>>	Transition	
	<<IntDec>>	Synch(H)	
	<<E_Action>>	Transition	
	<<T_External State>>	Transition	
	<<IntSig>>	Synch(H)	
	<<E_Signaling>>	Transition	

The *internal of DRA X* is represented by three different States: **IA_InitialState**, **C_DRA** and **External Objective**. The **<<IA_InitialState>>** is a state where an Internal Agent is in when it is first created; the **<<C_DRA>>** denotes the starting of Internal State of Agent Reactive Decisional. The **<<External Objective>>** represents the agent interpretation of each **<<T_Action>>**, and defines therefore its different behaviors; also, it delivers one or several **<<T_External States>>** to represent the objective state emitted to the environment.

The *External Objective* behavior is described with four states: *EO_InitialState*, *C_Internal State*, *Internal Objective*, and *Internal State*. The **<<EO_InitialState>>** is the one that an *External Objective* is in when it is first created; the **<<C_Internal State>>** denotes the starting of *Internal State* of all the *External Objective* and it generates the first **<<E_Decision>>** that represents a solution concerning process behavior in the future; it is characterized by its **Action Horizon AH(d)**, the time during which this decision remains valid. The **<<Internal Objective>>** (IO) represents the expected state after the execution of the decision. The transaction between IO and the others states is realized by trigger and effect, depending on each situation, the transaction could be realized in four types: AH (*Action Horizon*), AH/*External_State*, Signaling and Signaling/*External_State*. The **<<T_AH>>** only associated to *C_Internal State*, is generated when the deadline is finished. Moreover, the **<<T_AH/E_External_State>>** adds that when the duration is achieved the effect *External_State* is produced and it is associated to **<<Internal State>>**, which indicates the current state of the agent. Otherwise, if the time is respected, a signaling is generated, and reflects at any time the state of the controlled tools used to achieve a specific goal; in this case, there are two types of transactions, transaction with or without effect; the first is called **<<T_Signaling>>**, it is produced when there are one or more decisions in the future, the second is **<<T_Signaling/E_External_State>>**, it is the last transaction that produce the *External_State* effect, both of them are related to *Internal State*.

The *Communication Interface* (CI) is an abstract concept that represents the interaction between Agent Supervisor (AS) and the Sub-Agent, this interaction is realized with two types: IntDec (Decisional Interface) and IntSig (Signaling Interface). The <<IntDec>> (Top/Down) translates a decision (d) generated by the AS into several actions, each action is intended for a sub-agent of the lower level. The <<IntSig>> (Bottom/Up) synchronizes the external states (ES), sent by each sub-agent, and emits one signaling (s) intended for the AS. The modeling of CI is realized with different stereotypes. Once Internal Objective received the <<T_Decision>> effect, the IO interact with CI by sending the T_Decision d to the IntDec, which transform d into one or more <<E_Action>>, once the Agent analyze the E_Action and the time of decision was respected, the agent generates <<T_External State>> to the IntSig which synchronizes the ES and emits one <<E_Signaling>>. However, if the agent doesn't find a solution, it associates T_External State to exist state.

The metamodel that illustrates the Multi Reactive Decisional Agent System behaviors is presented in Fig. 3.

V. APPLICATION OF THE MDRA PROFILE

In disaster situations, the information plays an important role to solve and help victims if it is properly coordinated and responds to real needs. For this reason, the Rapid Assessment and Intervention Team (RAIT Team) is set up to react quickly with the event in order to accomplish an initial assessment helping to figure out the nature, scope of the incident and identify the required assistance, also, the performance of the RAIT Team is evaluated by the execution

time of the operations. So the RAIT Team can be considered as a Reactive System which has to coordinate both numerous actors and two or more reactive parallel sub-processes at a specific time. The RAIT Team is consisted of the following members: *Incident manager* (IM) provides the initial command centre, and supervises all team operations. *Logistics Specialist* (LS) evaluates the logistics need of medical supplies, food, water, etc. *Communications Specialist* (CS) institutes an emergency communications system, and determines long-term communications infrastructure requirements. *Planning Specialist* (PS) prepares a daily incident action plan, and develops a situation status system. *Medical Specialist* (MS) conducts a medical assessment, and identifies medical treatment requirements

The structure of the RAIT Team is modeled by using the first level of DRAS Profile, we consider the Incident Manager as an <<DRA Supervisor>> and the other specialties as an <<DRA Element>> (see Fig. 4). The internal behavior of each Agent is specified in each Package. In this example, we illustrate the “initiating Humanitarian Emergency”. (see Fig. 5-Fig. 7).

Fig. 3. Metmodel of MDRA profile behavior.

Fig. 4. Structure of RAIT team.

The rescue operation of Humanitarian Emergency is started when the Incident Manager receives a notification of an accident: `A1_IM_Initiate_Humanitarian_Emergency` (`A1_IM`). This notification enforces the IM to start the first stage of urgent actions to carry out: `D1_IM_first_Stage_Actions` (`D1_IM`). This decision is translated simultaneously on the corresponding agents (LS, CS, MS, PS), via the decision interface, as a set of specific actions respectively: `A1_LS_Evaluate_Logistics_Need`

(A1_LS), A1_CS_Get_first_Com_Needs (A1_CS), A1_MS_Assess_Med_Status (A1_MS) and A1_PS_Prepare_Daily_Action_Plan (A1_PS).

Fig. 5. Incident Manager specification.

Fig. 6. Logistics specialist specification.

Fig. 7. Communication specialist specification.

Once all of these actions are accomplished by the corresponding agent within the temporal limits (Action Horizon), the Signaling Interface aggregates the validation of all the External States into a single Signal: S1_IM_first_Stage_Ok (S1_IM).

The corresponding External States are:

- 1) ES1_LS_Logistics_Needs_Evaluated (ES1_LS)
- 2) ES1_CS_Emergency_Com_OK (ES1_CS)
- 3) ES1_MS_first_Med_Assess_Performed (ES1_MS)
- 4) ES1_PS_D_Action_Plan_Prepared (ES1_PS)

Once S1_IM is received, the IM agent starts the second stage of actions to achieve: D2_IM_second_Stage (D2_IM), the latter generates a set of actions received respectively by CS, MS and PS agents: A2_CS_Get_LT_Infras_Needs (A2_CS), A2_MS_Identify_Med_Treatment_Needs (A2_MS), and A2_PS_Set-up_Resources_Tracking (A2_PS).

When emitted, the corresponding External States (ES2_CS, ES2_MS, ES2_PS) are aggregated into a single Signal: S2_IM_second_Stage_Ok (S2_M), interpreted by the IM agent as a normal termination of the "initiating Humanitarian Emergency" process: emission of the External State ES1_IM_Initial_Stages_OK (ES1_IM).

VI. CONCLUSION

The contribution of this paper is to propose an UML Profile for modeling the internal and external behaviors of the Multi Decisional Reactive Agent System (MDRAS). The MDRA approaches based on Multi-Agent is devoted to specify and verify a reactive system. So, the MDRA profile objective is to improve the specification of such a system through modeling the structure of a system via package diagram, while a Statechart diagram models the internal and external interactions of the agent. Also, we have developed a plug-in accessible from the tool "Sparx Enterprise Architect" in order to make easier the MDRAS modeling. The works coming in the future will be converged on the examination of the use of other UML diagrams so as to extend the MDRA profile to cover other specificities of reactive system.

REFERENCES

- [1] R. A. Bermejo, "Reactive Operating System REactive Java Objects," *Electronic Journal on Networks and Distributed Processing*, no. 11, pp. 337–354, 2001.

- [2] R. J. Wieringa, *Design Methods for Reactive Systems: Yourdon, Statestate, and the UML*, 2003.
- [3] B. Bounabat, R. Romadi, and S. Labhalla, "Designing MultiAgent Reactive Systems: A Specification Method Based on Reactive Decisional Agents," H. Nakashima, C. Zhang (Eds.), *LNAI*, vol. 1733, Springer-Verlag Berlin Heidelberg, 1999, pp. 197-210.
- [4] A. Aaroud, S. E. Labhalla, and B. Bounabat, "Modelling the Handover function of Global System for Mobile Communication," *International Journal of Modelling and Simulation - ACTA Press*, vol. 25, no. 2, 2005.
- [5] R. Romadi, H. Berbia, and B. Bounabat, "Wireless Sensor Network Simulation of the Energy Consumption by a MultiAgents System," *Journal of Theoretical and Applied Information Technology*, vol. 25 no.1, pp. 50-56, March 15, 2011.
- [6] M. Berrada, "Qualitative Verification of Multi-Agents Reactive Decisional System Using Business Process Modeling Notation," *Intelligent Agent Technology*, pp. 747-751, 2006.
- [7] R. Cervenka, I. Trencansky, and M. Calisti, "Modeling Social Aspects of Multi-Agent Systems: The AML Approach," J. P. Muller and F. Zambonelli (Eds.), *AOSE 2005, LNCS*, vol. 3950, 2006, pp. 28-39.
- [8] G. T. A. Guedes and R. M. Vicari, "Applying a UML Profile in the Requirements Modeling to Multi-Agents Systems," presented at Seventh International Conference on the Quality of Information and Communications Technology, 2010.
- [9] P. Wongthongtham, D. Dillon, T. Dillon, and E. Chang, "Use of UML 2.1 to Model Multi-Agent Systems based on a Goal-driven Software Engineering Ontology," in *Proc. Fourth International Conference on Semantics, Knowledge and Grid*, 2008, pp. 428-432.
- [10] V. T. de Silva, R. C. Noya, and C. J. P. de Lucena, "Using the MAS-ML to model a multi-agent system," *Software engineering for multi-agent systems II: research issues and practical applications*, vol. 2940, 2004, pp. 129-148.
- [11] K. Hamilton and R. Miles, *Learning UML 2.0*, 2006.

Abdelhay Haqiq was born on April 5, 1986 in Rabat (Morocco). He received master degree in computer engineering from ENSIAS (National Higher School for Computer Science and System Analysis), Rabat, Morocco, 2010. Since 2011 he is a PhD candidate at ENSIAS, member of AL-QualSADI research team (Quality of Software Architectures, their Development and Integration). His field of interest includes information systems, UML Profile, Multi-Agent and formal specification and verification of reactive systems.

Bouchaib Bounabat was born on March 8, 1966 in Marrakesh (Morocco). He is a professor in ENSIAS, Rabat, Morocco. He received his master degree in computer science from Faculty of Science and Technology, Lille 1, French, 1989, Master of Advanced Studies in Engineering Sciences, Paris XII, 1990, and PhD in Computer Sciences from Institut National des Télécommunications, 1993. He is Responsible of "Computer Engineering" Formation and Research Unit in ENSIAS, International Expert in ICT Strategies and E-Government to several international organizations, Member of the board of Internet Society - Moroccan Chapter. His research interests include Quality of software architectures, Software Engineering Methodology and formalization, generic infrastructure development and integration.