Process Management Oriented Qualitative Verification for Multi-Agents Reactive Decisional Systems

Mohammed Berrada[1], Bouchaib Bounabat[2], and Mostafa Harti[3]
[1, 3] Mathematics&Computer Sciences Department
Faculty of Sciences, Sidi Med ben Abdellah University
B.P 1796 Atlas-Fès, 30000 Fez, Morocco
mohammed.berrada@gmail.com[1], mharti@fsdmfes.ac.ma[3]
[2] Al-Khawarizmi Computer Engineering Laboratory
Ecole Nationale d'Informatique et d'Analyse des Systèmes, ENSIAS
Rabat - Morocco
bounabat@ensias.ma

Abstract. A Reactive system is one that is in continual interaction with its environment, and executes at a pace determined by that environment. Due to their complex nature, reactive systems are extremely difficult to specify and validate. In this paper, we propose a new formal model for verification of such systems using Business Process Modeling Notation (BPMN). This approach considers a Reactive System as a Reactive Multi-Agent System consisting of concurrent reactive agents that cooperate with each other to achieve the desired functionality. BPMN is used as a verification tool in order to verify the systems behaviors. Finally an example of an application of the approach, and its associated support tool, is mentioned.

Keywords : Agent, BPMN, Organizational system, Qualitative verification, Reactive systems.

(Received December 20, 2006 / Accepted July 13, 2007)

1. INTRODUCTION

A Reactive system is one that is in continual interaction with its environment, and executes at a pace determined by that environment. Examples of such systems are network protocols, industrial-process control systems, organizational systems, etc. Reactive systems are responsive systems consisting of two or more reactive parallel sub-processes that continuously cooperate to achieve a pre-defined goal [17]. Such systems are intrinsically state based, and transition from one state to another is based on external and internal events. Thus, Reactive systems are complex computer systems, and may not be modeled by transformational techniques.

The use of rigorous formal methods in specification and validation can help designers to limit the introduction of potentially faulty components during the construction of the system. Verification is an important stage in reactive system design where the designers specify the desired behavior and determine if this system is successful. One approach for validation is the qualitative verification, which considers observable behavior as criteria to determine success.

Due to their complex nature [20], reactive systems are extremely difficult to specify and validate. In this paper, we propose a new formal model for the specification and the validation of such systems. This approach considers a Reactive System as a Multi-Agent Reactive System, i.e. a distributed system consisting of several autonomous reactive agents that coordinate their action in order to fulfill usually joint but also sometimes competitive tasks. Concurrency is further characterized by the need to express communication and synchronization among concurrent agents. In another hand, the proposed approach uses the business modeling techniques for the qualitative verification for such systems, especially when they are applied to model organizational systems.

This paper is organized as follows: Section 2 sets out the Multi-Agent Reactive Decisional System model and its verification aspects. Section 3, which presents the concepts of this new approach, describes the use of BPMN as a verification tool of the Multi-Agents Reactive Decisional System behavior under normal and degraded functioning. Section 4 presents an example that illustrates the application of this approach in the organizational modeling domain.

2. DESCRIPTION OF MARDS MODEL

The Multi-Agent Reactive Decisional System (MARDS) [1]-[2]-[11]-[12] constitutes an approach among the newest and most useful ones for reactive system modeling. Its main components are Decisional Reactive Agents (DRA), which are interconnected with them by communication interfaces.
In this section, we are going to define, formally, all components constituting a MARDS.

2.1 Decisional Reactive Agent model

A DRA (Decisional Reactive Agent) can be used to define an autonomous and independent agent [25]. The obtained agent receives actions and can act in an autonomous way until their realization in the adequate deadlines [24]-[25]. To reach a given goal, we need to define an objective or a sequence of objectives ordered in a certain proposed way, which we propose in order to resolve this problem.

2.1.1 General architecture of a DRA

A DRA is a reactive agent model developed by Bounabat [11]. Its functional architecture bases itself essentially on the decision-making organ introduced into ADMDOOS [10] (Analysis and Design Method of Decisional Oriented Object Systems).
The DRA Concept was applied in modeling and the checking of the automated systems of production as in the mobile systems domain [3]-[4]-[5]. This work consists in applying this concept in the domain of the organizational systems modeling (company, Information system Department...).
Figure 1 shows the internal structure of a reactive agent, adopted in the case of an agent of DRA type.

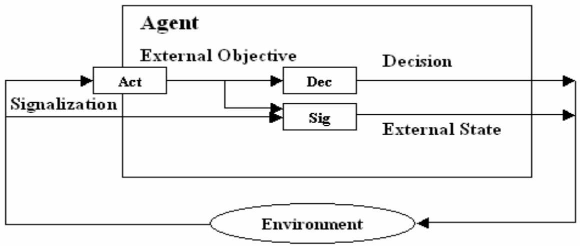


Figure 1: The perception -deliberation - execution structure of a DRA.

The functions Dec and Sig perform the role of execution and deliberation; they aim at emitting messages (decision and external state) toward the outside world (environment) once an action is perceived by the Act function.

2.1.2 Formal description of a DRA

The formal description of a DRA consists of the formalization of the various constituents of a decisional agent. A decisional reactive agent is a 10 uplets Ag = <A, D, S, E ', O, E, O', Act, Dec, Sig>, where:

- A: Set of actions exerted on the agent. Each action, undergone by an object, represents a possible operation to be carried out on this object in order to achieve a specific goal.
- D: Set of decisions generated by the agent. Each decision is a solution concerning process behavior in the future; each decision is characterized by its action horizon: ***Ha***, the time during which this decision remains valid.
- S: Set of Signalling received by the agent. Each Signalling received by an object, reflects at any given time the state of the controlled tools used to achieve a specific goal.
- E': Set of external states delivered by the agent. Each one represents the object state emitted to the environment.
- E: Set of agent's internal states. Each one indicates the current state of the agent.
- O: Set of agent's internal objectives. Each decision is elaborated in order to achieve an internal objective according to the current external objective and the actual internal state.
- O': Set of agent's external objectives which can be achieved. These objectives represent the agent's interpreting of each action.
- The function Act interprets an action as an external objective that it used respectively by the functions Dec and Sig to generate respectively the appropriate responses: (decision, internal objective) and (internal state, external state).

For a DRA, the events that come from (respectively send toward) the environment can be only actions or signalizations (respectively decisions or external states).

2.2 MARDS model

A Multi-Agent Reactive Decisional System (MARDS) [12] is a software structure characterized by a set of agents, interconnected by communication interfaces.

2.2.1 General architecture of a MARDS

The internal structure of a MARDS is based on a two-level tree (Figure 2), consisting of DRA Supervisor (DRAS), two or several possible sub-agent components (MARDSi) and communication interfaces (Decisional Interface and Signalization Interface) that interconnect the supervisor with its sub-agents.

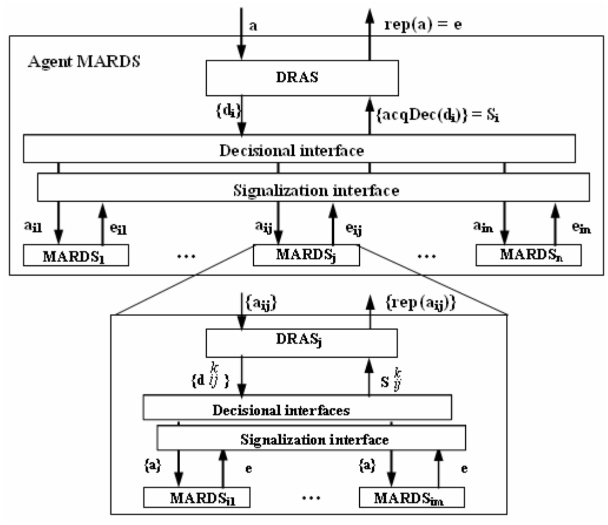


Figure 2: Internal Structure of a MARDS
A MARDSi can be either a simple DRA or an MARDS.

2.2.2 Formal description of a MARDS:

The description of a MARDS refers simply to the formalization of the basis components described in figure 2. The DRA component is described in the previous section. Now, we are going to define, formally, the decisional and the signalization interfaces.

Formalization of the decisional interface:

Formally, a decisional interface is an uplet DI =<DI_Input, n, DI_Output, TradDec>, where:

- DI_Input : inputs of DI (decision)
- n : dimension of DI (the number of channels that constitute the outputs of DI),
- DI_Output : outputs of DI (set of actions),
- TradDec : translation function of a decision into several parallel actions, each of these actions is led to an inferior sub-agent level.

Formalization of the signalization Interface:

Formally, a signalization interface is an uplet SI =<SI_Input, n, SI_Output, TradSig> where:

- SI_Input : inputs of SI (Set of external states),
- n : dimension of SI (the number of channels that constitute the inputs of SI),
- SI_Output : outputs of SI (signalization),
- TradSig : the translation function of several external states into one and only one signalization.

A Mutli-Agent Decisional Reactive System (MARDS) can be defined then as a 4-uplet S: <DRAS, DI, SI, E-MADRS>, where:

- DRAS : An agent of DRA type that supervises S,
- DI : Decisional Interface of S,
- SI : Signalling Interface of S,
- E-MADRS: Set of MARDS components S. It can be two or more.

2.3 MARDS sub-agents components verification

The formal verification consists of comparing a description of a system with the set of properties which it must adhere to. Initially, system must be described in a language modeling that has a well defined operational semantics. This will make it possible to associate any given system with the labeled transitional system (i.e., a set of states provided with a transitional relation) which models its behavior (i.e., a set of all the possible executions).

2.3.1 A critical review of reactive agent verification tools

Among the few current approaches for specifying Reactive Agents: [19] describes agents, tasks and environments using the Z specification language [23]; [16] specifies the reactive agent behavior by Real Time Knowledge models; [7] describes agent using temporal logic tools.

In spite of the diversity of the used theories, two classes can be distinguished: qualitative (state-transition orientated models), and quantitative (logic verification orientated models) whereby the formalism is taken into account to express and check the temporal constraints [18]-[21].

The association of these two techniques, qualitative and quantitative, is thus necessary in order to make any checking as complete as possible.

2.3.2 MARDS quantitative verification

Principle of quantitative verification:

The quantitative verification makes it possible to characterize the total properties of the program such as the absence of blocking. The properties to be checked are generally expressed by temporal logic formulae which are interpreted on the set of the labeled transition systems. The specification is thus represented by an infinite class of labeled transition systems and the checking ensures that the particular system belongs to this class.

MARDS Verification with RTTL:

Real Time temporal Logic (RTTL) has been widely used for the specification and verification of concurrent systems [22]-[25]-[26], especially of Multi-Agent Reactive Decisional Systems [1]-[2].

2.3.3 MARDS qualitative verification

Principle of qualitative verification:

This type of verification consists of the description of the behavior of the program observed at a certain level of abstraction. The properties are, therefore, represented by a labeled transition system which could result from a program expressed in a language that the program has already checked. The comparison between the two systems of labeled transitions is carried out by means of a relation of equivalence; the choice of which is a function of the criteria of abstraction that is desired to take into account.

MARDS Verification with ESTEREL:

ESTEREL is a synchronous language, whose principle is based on the reduction of automats [8]-[9]-[13] was already used to check the qualitative way of the MARDS behavior describing a system subjected to strong temporal constraints (example: GSM systems [3]-[4]-[5].).

BPMN based MARDS qualitative Verification:

The use of such tools proves to be inadequate when it is a question of checking the behavior of an organizational system (example: company, information systems department or even a team). Such systems, where each element can be represented by a DRA, can be described as MARDS. Their evaluation consists in:

- Checking the relationships existing between various hierarchical levels (Action → Decisions) contributing to the achievement of the aforementioned objectives.
- Observing the reaction of the set of the organization in case of failure or a risk occurring in one of the hierarchical levels.

Business Process Modeling Notation [14]-[15], because of its representation of the actors and the activities, is completely made adequate to allow such a checking. BPMN can be then used as a verification tool of the MARDS behavior under normal and degraded functioning.

3. BPMN AND MARDS

BPMN stands for Business Process Modeling Notation [6]-[14]-[15]-[27]. It is the new standard for modeling business processes and web service processes, as put forth by the Business Process Management Initiative (BPMI). BPMN is a core enabler of a new initiative in the Enterprise Architecture world called Business Process Management (BPM).

3.1 BPMN description of DRA external behavior

This section presents the representations rules of DRA concepts using BPMN (Table 1):

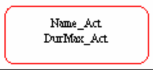
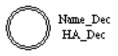
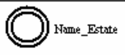
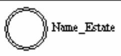
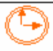
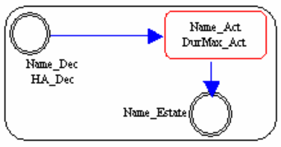
DRA Components	Correspondent BPMN element	
Action	Task	
Decision	Intermediate Event	
External State	End Event if it is the last state	
	Intermediate Event to trigger the next decision	
HA surpassed	Timer Event	
Sequence Flow	Sequence Flow	
External Objective	Sub-Process	

Table 1: Representation of the DRA's components

For the sub-process, when receiving a decision the concerned DRA agent starts an action, and with the case of a normal operation, it will be able to emit the adequate external state

3.2 Degraded functioning

3.2.1 Degraded functioning principle

In order to palliate to the lag problems or the malfunctioning of certain entities, we have added this degraded functioning at the level of DRA specification. The objective is to describe how the system can operate even in the presence of errors, without the loss of functionalities or performance.

This principle can be interpreted as if the agent is able to "delay" the deadlines waiting for the decision acknowledgement. These delays can be shown to the environment by sending a specific external state.

If a decision surpasses its related action horizon, a graceful degradation operating mode will be automatically trigged. The Degraded External Objective DextO generates new decisions and that will not necessary have the same external state of the starting action.

In change, the amount actions horizons associated to the decisions generated by this new objective, will be added to the maximum duration warned initially (necessary time for the accomplishment of the action). Then the function DurMax (associating the External Objective extO to the longest duration of its operations execution.)

of this action will be under the following form:

$$\mathbf{DurMax\ (act)} = \begin{array}{c}\sum \mathbf{HA(decisions\ generated\ by\ extO)} \\ + \\ \sum \mathbf{HA(decisions\ generated\ by\ DextO)}\end{array} \qquad \mathbf{(1)}$$

3.2.2 Degraded functioning modeling

Even a decision has surpassed its initial action Horizon (Figure 3), the system must continue to operate accepting a partial degradation of functionally of performance during recovery or repair.

Thus, when *Dec1* surpasses its action horizon (*Ha1*), the initial external objective (*Normal functioning*) will be stopped while a second external objective (*Degraded functioning*) will be started.

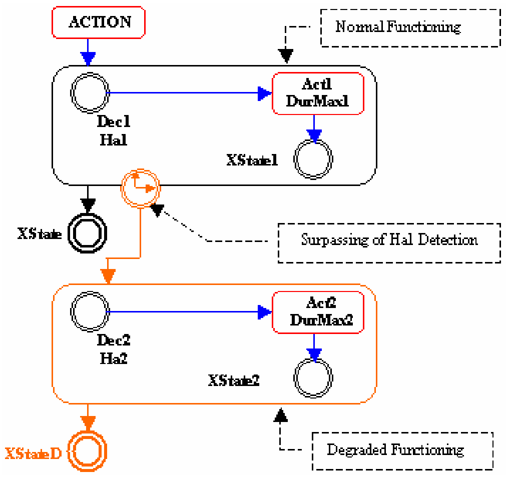


Figure 3: Degraded functioning Modeling

3.3 BPMN description of MARDS behavior

The objective here is to model an MARDS system (Figure 4) using BPMN.

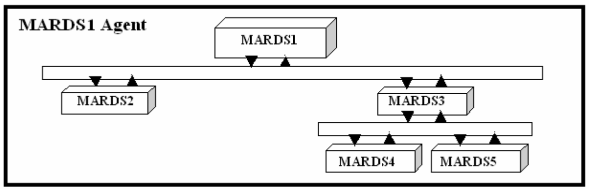


Figure 4: MARDS1 agent

The figures 5 and 6 display respectively the BPMN models of MARDS1 agent and MARDS3 sub-agent. These models are generated by the verification interface (section 4.3) of the associated tool of this proposed approach.

The action *Act1* received by *MARDS1* corresponds to an external objective (*ExtObj*) which generates decisions {*Dec1.1, Dec1.2*}. Each decision corresponds to a several actions received by MARDS2 {(*Dec1.1*, *Act2.1*), (*Dec1.2*, *Act2.2*)} and by MARDS3 {(*Dec1.2*, *Act3*)}.

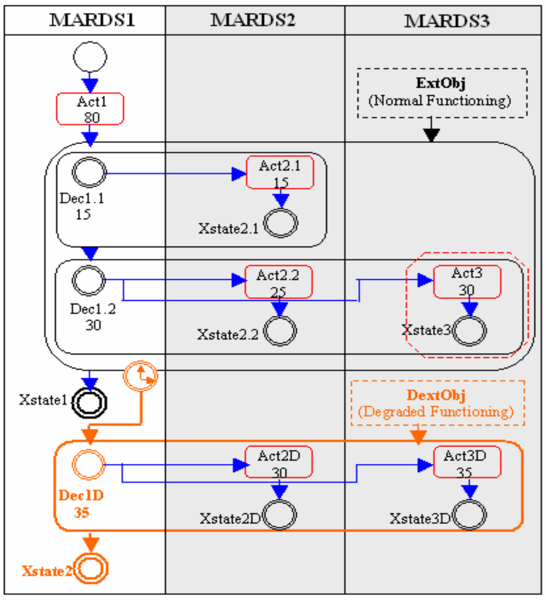


Figure 5: BMPN Model of MARDS1 agent

The actions (*Act2.1, Act2.2*) are just simple tasks but the *Act3* is an activity because it's received by the composed agent *MARDS3* and it's modeled as a sub-process

The degraded functioning of *MARDS1* is described by the external objective (DextObj) which is started when one of the decisions (*Dec1.1*, *Dec1.2*) has exceeded its action horizon.

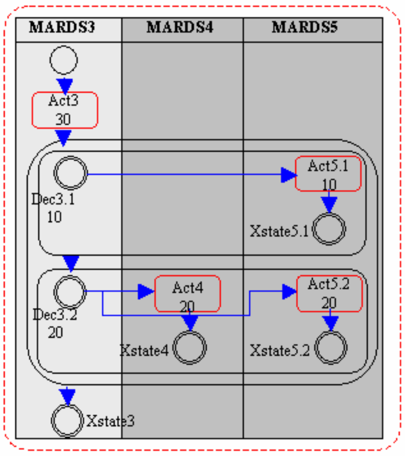


Figure 6: BPMN model of MARDS3 sub-agent

The modeled sub-process displayed in figure 6 presents

the (*Dec1.2, Act3*) sub-action received by *MARDS3*.
All sub-processes realized by the appropriate sub-agents compose the principal process (Figure 5) trigged by the initial action called *Act1*.

4. Example: Information Technology (IT) Strategic Plan Process Modeling

4.1 General description

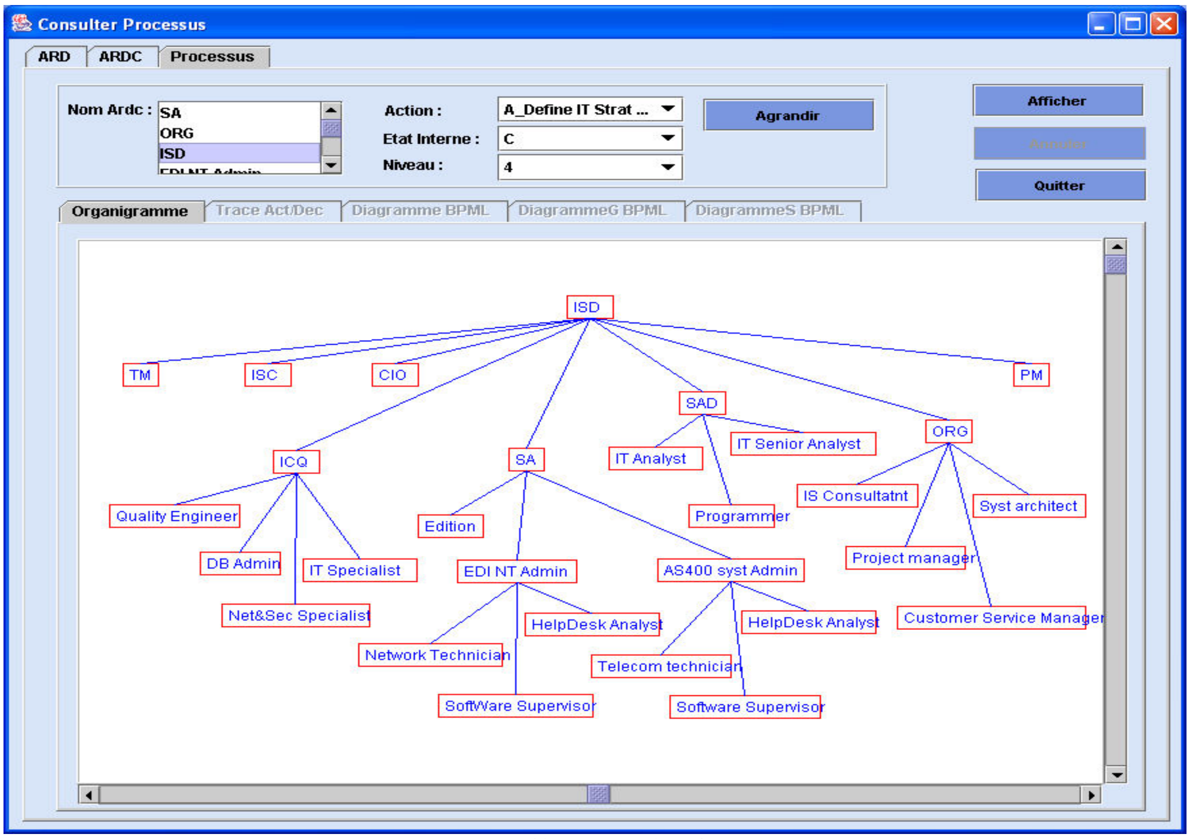
The objective here is to describe and to evaluate in a qualitative way, the process of building the IT Strategic Plan (ITSP) of an Information Systems Department (ISD) of a company.
ITSP has an objective to obtain a lining document that the Chief Information Officer (CIO) must revisit annually to identify new initiatives or refer existing ones that continue during the ISD capacities.
Several poles can compose the ISD:

SAD: Software Application Development.
SA: Systems (OS, Network, Databases …) Administration.
PM: Project Management.
ICQ: Insurance and Control Quality.
CIO: Chief Information Officer.
TM: Top Management.
ORG: Organization.
ISC: Information System Committee.

In the sequence of this example, we consider each component or entity as MARDS including the ISD.

4.2 System modeling using the support tool

Figure 7 describes the overall presentation of the hierarchical system using the PRV (**P**rocess Notation based MA**R**DS **V**erification) tool where each ISD pole is constituted by many sub-poles and by other simple entities. The DSI agent components can be MARDS systems (ICQ, SAD, AS400 system Admin …) or simple DRA agents (CIO, Quality Engineer, Network Technician …).


Consulter Processus

Nom Ardc : SA, ORG, ISD, EDI NT Admin
Action : A_Define IT Strat ...
Etat Interne : C
Niveau : 4

Figure 7: Graphic representation of ISD agent

4.3 Verification Interface

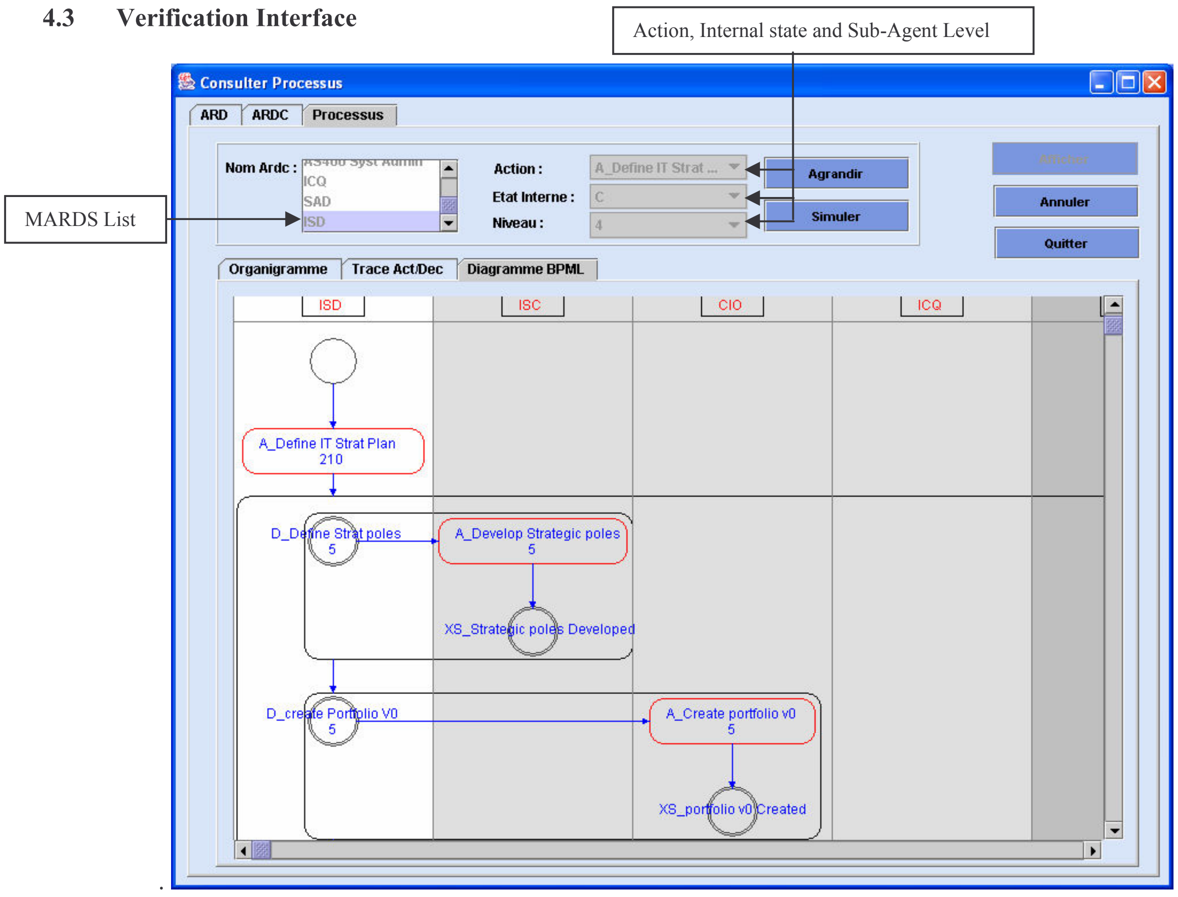


Figure 8: Verification Interface

This interface allows designers to specify and verify the MARDS behavior by using BPMN.

The specification stage can be realized by modeling the set of sub-processes that composes the process trigged. For every MARDS system, we can trigger many processes in line with the parameters (*action* and *internal state*) chosen initially. The parameter called *Level* characterizes the level of hierarchical display of modeled system agents. Therefore, the verification of the behavior of modeled system can be achieved if any contributing agent in the principal process respects his time constraints. This also can be obtained by the transition to degraded functioning mode of an agent that can exceed any action horizon decision.

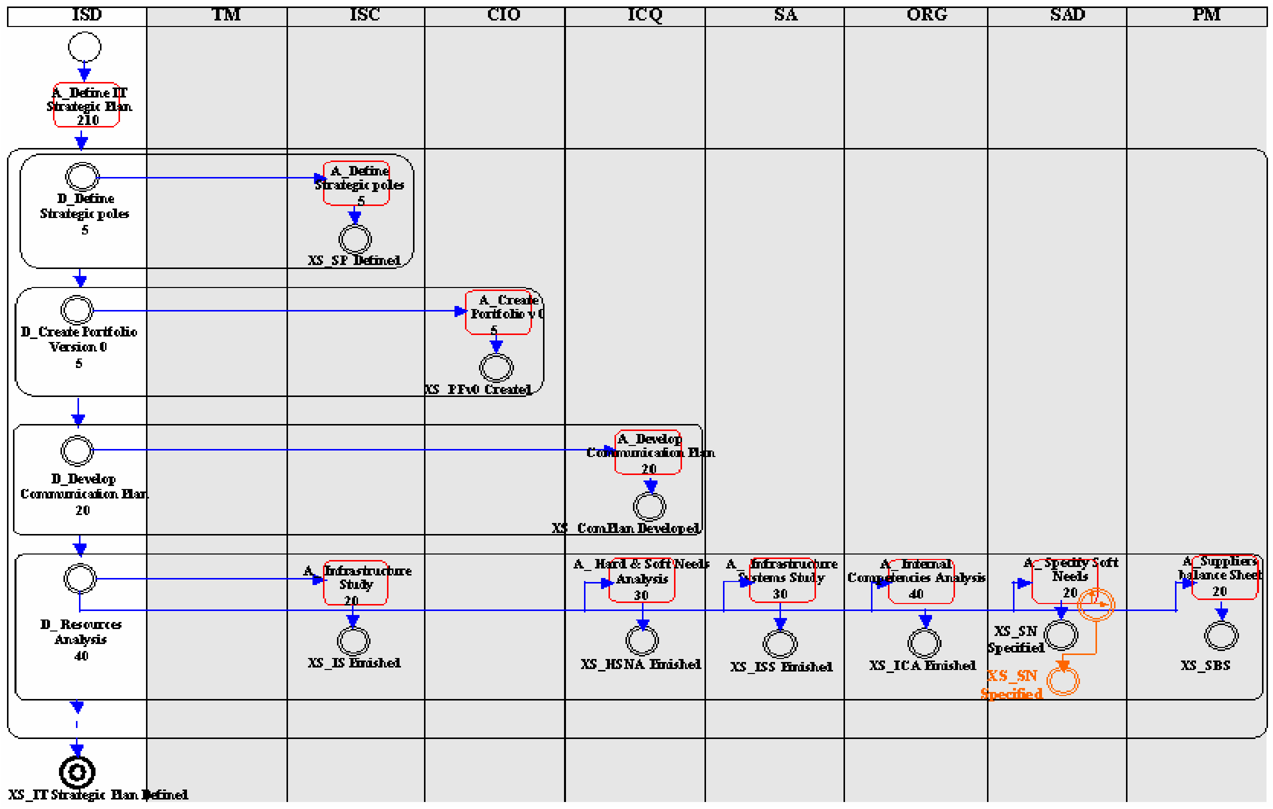
The action « *A_Define IT strategic plan* » which can be transformed as a sequence of many decisions (D_Define Strategic poles, D_Create Portfolio v0 …) has been chosen for ITSP process. Every decision represents a sub-process since it generates new actions corresponding to agents at lower level (ISC, CIO, ICQ…). For example, « *D_Define strategic poles* » decision generates the « *A_Develop strategic poles* » action to *ISC* agent; the end of this action triggers the second « *D_Create Portfolio v0* » decision to *DSI* agent. This carries on until the achievement of all sub-processes and automatically the end of ITSP process.

4.4 Results interpretation

Figure 9 represents ITSP model in the second level in system hierarchy. This model is composed by many sub-processes, every one describes a decision generated by the starting action called « *A_Define IT Strategic plan* ». Among sub-processes shown in this model are « Define strategic poles », « Develop communication plan » and « resources analysis». The first and the second ones are simple sub-processes because they both generate at the same time one decision for any one agent.

Figure 10 details the last sub-process «resources analysis» in the third level in system hierarchy. The degraded functioning can be started by « *A_Specify Software Needs* » action generated by *SAD* agent. So, if the first decision « *D_Specify SN*» (SN: Software Needs) generated by *IT Analyst* agent, has exceeded its action horizon, then another decision (same name) will be generated by *IT Senior Analyst* agent. These two decisions have the same external state

« *XS_SN Specified*» and the sum of their action horizons is no longer than the maximum duration for achieving the initial action.
Thus, verifying the modeled process (Figure 9), the ITSP can check easily if each entity of the department contributes effectively to the building of the IT action plan and if the time constraints expressed (here actions horizons) can be respected.


ISD
TM
ISC
CIO
ICQ
SA
ORG
SAD
PM
A_Define IT Strategic Plan 210
D_Define Strategic poles 5
A_Define Strategic poles 5
XS_SP Defined
D_Create Portfolio Version 0 5
A_Create Portfolio v 0 5
XS_PFv0 Created
D_Develop Communication Plan 20
A_Develop Communication Plan 20
XS_ComPlan Developed
D_Resources Analysis 40
A_Infrastructure Study 20
XS_IS Finished
A_Hard & Soft Needs Analysis 30
XS_HSNA Finished
A_Infrastructure Systems Study 30
XS_ISS Finished
A_Internal Competencies Analysis 40
XS_ICA Finished
A_Specify Soft Needs 20
XS_SN Specified
XS_SN Specified
A_Suppliers balance Sheet 20
XS_SBS
XS_IT Strategic Plan Defined
Figure 9: IT Strategic Plan Process Model

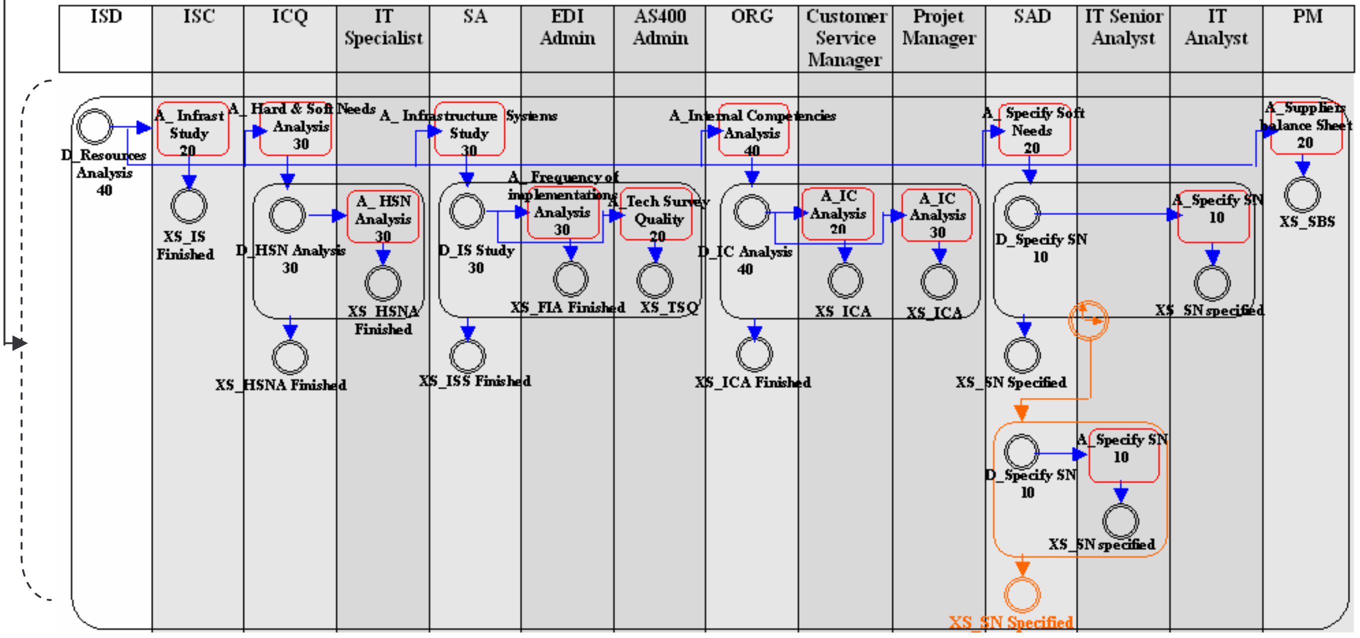


Figure 10: Resources analysis sub-Process Model

5. CONCLUSION

The contribution of this paper is to give a new formal approach to deal with description and verification of an organizational reactive system. The originality is to consider each component of reactive system as a Reactive Decisional Agent. With its top-down process and its principles of decomposition, this method allows to get a model which is more easily understandable by the designer of such systems. The BPMN is used here in order to check the reactive agent behaviors in a qualitative way. The mechanism of action horizon, the time during which an agent decision remains valid, is moreover useful to specify temporal performances, and then to detect abnormal functioning modes. The approach proposes a fault tolerance techniques dealing with such problems, and is supported by an automated tool of Verification.

A lot of work can be done in this area. We want to study how BPEL4WS [6] can be used in the automatic simulation of BPMN models generated by the PRV tool, with potentially large rewards in making web services a business reality.

6. REFERENCES

[1] Aaroud A, Labhalla S.E, Bounabat B. *Designing Multi-Agent Reactive Systems a specification method based on Decisional Reactive Agent*. New Computing Group, February, 2001.

[2] Aaroud A, Labhalla S.E, Bounabat B. *A New Formal Approach for the Specification and the Verification of Multi-Agent Reactive System Operating Modes*. International Journal for Information Processing and Technology. Nova Sciences Publishers, Inc March, 2001.

[3] Aaroud A, Labhalla S.E, Bounabat B. Application de l'approche SMARD pour la modélisation de la fonction handover du système GSM " Actes de conférence SETIT, Sousse-Tunisie, 17-21 Mars, 2003 ;

[4] Aaroud A, Labhalla S.E, Bounabat B. *An Agent Based Approach to design a GSM location Update*. IEEE Conf. SoftCom-International Conference on Software. Telecommunication and Computer Networks, 10-13 October, 2004.

[5] Aaroud A, Labhalla S.E, Bounabat B. *Modelling the Handover function of Global System for Mobile Communication*. in the International Journal of Modelling and Simulation- ACTA Press/ Vol 25, N° 2, 2005.

[6] Andrews T, Curbera F, Dholakia H, Goland Y, Klein J, Leymann F, Liu K, Roller D, Smith D, Thatte S, Trickovic I, and Weerawanara S. *Business Process Execution Language for Web Services Version 1.1*. Technical report, BEA Systems, IBM, Microsoft, SAP, Siebel, 5 May, 2003. http://www.ibm.com/developerworks/library/ws-bpel/

[7] Bahssoun J.P, Merz S, and Servieres C. *A Framework for formalizing and proving concurrent objects*. WS4ECOOP'94, Bologna Italy, July 1994.

[8] Berry G, Couronne P. *Synchronous programming of reactive systems: an introduction to ESTEREL*. IEEE Software Engineering, Vol.16, No.4, 1987.

[9] Berry G. *The ESTEREL V5 Language Primer*. Internal Report, CMA Ecoles des Mine, INRIA, Paris, 17 Mars, 1998.

[10] Bounabat B. *MACSOOD, Méthode d'Analyse et de Conception des Systèmes Orientés Objet Décisionnel, Application à la conception des Systèmes Automatisés de Production*. Doctoraldiss, Evry institute of telecommunication, France, September, 1993.

[11] Bounabat B, Romadi R, Labhalla S.E. *User's behavioural requirements specification for a reactive agent*. IEEE publication-CESA'98, Nabeul-Hammamet, Tunisia, 01/04 April, 1998.

[12] Bounabat B. *Méthode d'Analyse et de Conception Orientée Objet Décisionnel. Application aux langages synchrones et aux systèmes répartis*. doctoraldiss, Cadi Ayyad University, Faculty of sciences, Marrakech, Morocco, 2000.

[13] Boussinot F, and R. de Simone. *The ESTEREL language*. Proceeding of the of the IEEE, Vol.79, No.9, pp:1293-1304, 1991.

[14] BPMI, *Business Process Modeling Notation (BPMN) Specification Version 1.0*. May 3, 2004. http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf

[15] BPMI, *Business Process Modeling Notation Specification, OMG Final Adopted Specification*. February 6, 2006. http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf

[16] Charpillet F, Boyer A. *Incorporating AI Techniques into Predictable Realtime Systems*. Quatorzièmes Journées Internationales d'Avignon, 1994.

[17] Furbach. *Formal specification methods for reactive systems, Journal of Systems Software*. No. 21, pp. 129-139, 1993.

[18] Gasnier F. *Modélisation structurelle du temps par Réseaux de Petri: une contribution à la validation temporelle*. Thèse d'université, Ecole Centrale de Nantes, Novembre, 1998.

[19] Goodwin R., Formalizing Propreties of Agents, CMU Reports, CMUCS93159, 1993.

[20] Harel D, Lachover H, Naamad A, Pnueli A, Politi M, Sherman R, Shtul-Trauring A. *STATEMATE: A Working Environment for the Development of Complex Reactive Systems*. Proceedings of the 10th international conference on Software Engineering, Singapore, April 11-15, 1988.

[21] Kappellos K. *Environnement de programmation des applications réactives*. Thèse d'université, Ecole des Mines de Paris, 7 Novembre, 1994.

[22] Lamport L. *What Good is Temporal Logic?*. Proceeding of IFIP, Information Processing, pp. 657-668, 1983.

[23] Mataga P.A, Zave P. *Formal specification of Telephone Features*. Z User Workshop, Bowen & Hall, eds., pp. 2950, Cambridge, SpringerVerlag, 1994.

[24] Mensch J.S, Kersual D, Crespo A, Charpillet F, Pessi E. *Reakt, Real Time Architecture for Time critical knowledge-based system*. Intelligent Systems Engineering, 1994.

[25] Ostroff, J.S. *Temporal Logic for RealTime systems*. (Advanced Software Development Series. Research) 1st edn, (1989).

[26] Pnuelli A. The *Temporal Semantics of Concurrent Programs*. Theoretical Computer Science, No.13, pp. 45-60, 1981.

[27] Stephen A. White - IBM, *Workflow Patterns with BPMN and UML*. January, 2004. http://www.bpmn.org/Documents/Notations%20and%20Workflow%20Patterns.pdf