Business Modeling of Enterprise Architecture Based on Multi-Agent System

Mohammed Berrada and Bouchaib Bounabat

Abstract—**The enterprise architecture (EA) purpose is to provide a clear and comprehensive picture of the structure of an entity, either an organizational or a functional or a mission area. It is an essential tool for effectively and efficiently engineering business processes and for implementing and evolving supporting systems. The methods, approaches, languages for developing and describing EA, such as The Open Group Architecture Framework, acknowledge the importance of business modeling in the development of EAs. Business modeling support is needed to specify and document EA components and their relationships. Multi-agent Systems (MAS) are composed of autonomous agents that interact and coordinate to achieve their intentions. This makes them particularly adapted to modeling complex information systems composed of both human and software actors. In this paper we propose a new approach for EA Business modeling using two concepts: (1) The Multi-Agent Reactive Decisional System (MARDS) Model for specifying and implementing the EA components and their relationships; and (2) the Business Process Management (BPM) concepts for modeling and simulation of EA processes.**

Index Terms—**Business modeling, enterprise architecture, enterprise architecture modeling, multi-agent system.**

I. Introduction

Whether economic conditions are challenging or favorable, organizations should always seek to maximize the value from their IT investments, identify and remove waste and ensure that future investment decisions are based on what the business needs in the long term (strategic) as well as the short term (tactical). Developing and maintaining a model of the EA has become an established approach to achieving these objectives.

Enterprise architecture describes the enterprise structure. It represents all aggregate artifacts that are relevant to a company. There are many frameworks used to describe enterprise architecture such as [1]-[3] etc. But, it is often modeled as a layered organization. The layers that are usually recognized are the business layer, the application layer, the information layer and the technology layer.

MAS are composed of autonomous agents that interact and coordinate to achieve their intentions. This makes them particularly adapted to modeling complex information systems, such as EA, composed of both human and software actors [4]. This paper introduces a new way of modeling the EA, based on a MARDS model. It is simple yet powerful in nature, can be adopted incrementally, and provides tangible benefits to organizations seeking to understand the current state of their business and IT or implement strategy through projects.

The layout of this paper is as follows. The second section is devoted to the MARDS Model and its modeling aspects; the third section discusses the EA concept and presents the layered EA and its modeling benefits. Finally, an example is presented in the fourth section to apply and explain the concepts of the EA business modeling approach. The conclusion and future work are presented in Section V.

II. MARDS Model

A. MARDS Overview

A Multi-Agent Reactive Decisional System (MARDS) [5] is a software structure characterized by a set of Decisional Reactive Agents (DRA), interconnected by communication interfaces.

The MARDS Concept was applied firstly in modeling and the checking of the automated systems of production as in the mobile systems domain [6], [7].

In the second time, we have used, in [8]-[9], the MARDS model, and Business Process Management (BPM) languages, for modeling, verification and simulation of the behavior of organizational system (e.g. company, information system department or even a team). Such system, where each element can be represented by a DRA, can be described by MARDS. To support this work, we have developed, in [10], [11], the PAMS "BPM norms based MARDS Modeling and Simulation" tool that allows managers and architects to define agents, compose systems and generate business models and BPEL (Business Process Execution Language) processes of business processes achieved in organizational systems.

In the following sections, we are going to describe the general architectures and the business modeling concept of DRA and MARDS Models.

B. General Architecture of a DRA

Figure 1 shows the internal structure 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.

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:

Manuscript received May 6, 2013; revised July 28, 2013

Mohammed Berrada is with the ENSAF (National School of Applied Sciences of Fez), LSIS Laboratory, Sidi Mohamed ben Abdellah University, Fez, Morocco (email: mohammed-berrada@usmba.ac.ma).

Bouchaib Bounabat is with the ENSIAS, Al-Qualsadi Research & Development Team, Université Mohammed V – Souissi Rabat, Morocco (email: bounabat@ensias.ma).

DOI: 10.7763/IJEEEE.2013.V3.281

- A: Set of actions performed on Ag,
- D: Set of decisions generated by Ag,
- S: Set of signalization received by Ag,
- E ': Set of external states delivered by Ag,
- O: Set of internal objectives of Ag,
- E: Set of internal states of Ag,
- O': Set of external objectives of Ag,
- Act: Function that interprets an action performed on Ag in one external objective,
- Dec: Function that describes the agent reaction by generating a decision toward its environment,
- Sig: Function that receives a signalization and generates an external state toward its environment.

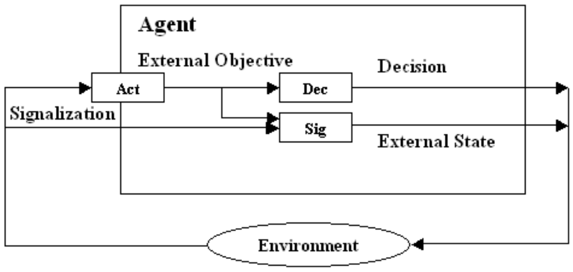

Fig. 1. The perception -deliberation - execution structure of a DRA.

For a DRA, the events that come from (respectively send toward) the environment can be only actions or signalizations (respectively decisions or external states). The external or internal objectives as well as the internal states constitute internal events. Each decision is characterized by its action horizon: ***Ha***, the time during which this decision remains valid.

C. General Architecture of a MARDS

The internal structure of a MARDS is based on a two-level tree (Fig. 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.

The description of a MARDS refers simply to the formalization of the basic components described in Fig. 2.

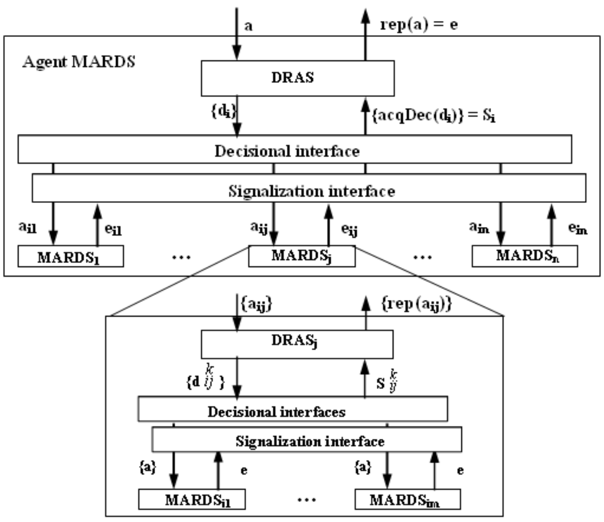

Fig. 2. Internal Structure of a MARDS.

A MARDSi can be either a simple DRA or an MARDS. A MARDS can be defined as a quadruplet S : <DRAS, DI, SI, E-MADRS>, where:

- DRAS: An agent of DRA type that supervises S.
- DI: Decisional Interface of S. This interface implements a translation function of a decision into several parallel actions; each of these actions is led to an inferior sub-agent level.
- SI: Signaling Interface of S. This interface implements a translation function of several external states into one and only one signalization.
- E-MADRS: Set of MARDS components S. It can be two or more.

D. Business Modeling of MARDS

This section presents the structure of processes achieved within a MARDS based on his internal architecture, and describes the principle applied for business modeling of these processes using the Business Process Modeling Notation (BPMN) [12].

1) Presentation of processes achieved within a MARDS

According to Fig. 2, the DRAS supervisor generates several decisions {d_i, i=1...m} in receiving the depart action {a}. Every decision is translated, by decisional interface, to one or several sub-actions {ai$k$, k=1... n} corresponding to appropriate agents {MARDSk}. The delivered external states, by these sub-agents, are received by the signalization interface and translated to one signalization Si. This signalization is the acquittal of the decision {d_i} which the supervisor either generates the next decision {d_{i+1}} or the end external state {e}, MARDS considers this external state as a response of the depart action. The sequencing of decisions is a process achieved by MARDS and generated by the depart action.

2) BPMN description of MARDS

The BPMN model displayed in Fig. 3 presents a prototype model for any MARDS [10]. The depart action, translated to the external objective ExtObj, generates several decisions {dec_i, i=...m} and every one must perform at least a sub-action act_ki corresponding to MARDSk sub-agent.

Every sub-action act_ki achieved by a MARDSk sub-agent is modeled as: a simple task if MARDSk is a simple agent; or a sub-process if MARDSk is a composite agent.

In this BPMN model, we have presented only the Normal functioning (ExtObj) of MARDS that will be used for the business modeling of EA. Another degraded Functioning mode has been developed in [8], in order to palliate to the malfunctioning of certain MARDSk agents.

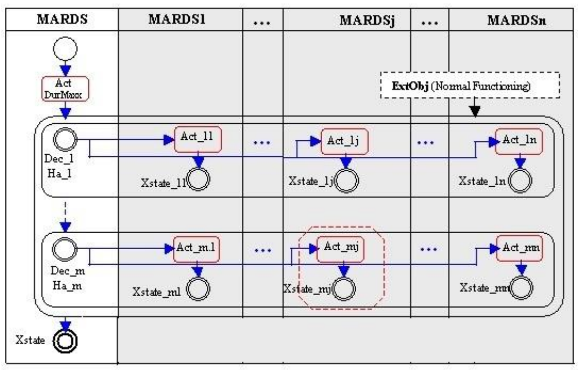

Fig. 3. BPMN model of MARDS agent.

III. Enterprise Architecture Overview

A. Enterprise and Framework Architectures: Concept Definitions

Definitions of EA have been developed and published by many industry experts and organizations involved with developing, promoting or supporting Enterprise architecture efforts.

One approach that has grown in popularity in the past few years is based on a framework developed by John Zachman. Zachman defines EA as "A comprehensive definition of Enterprise Architecture – A framework for improving enterprise communications about architecture issues." [1]. Thus, he identified the need to use a logical construction blueprint (i.e., architecture) for defining and controlling the integration of systems and their components.

The "2001 Practical Guide to the Federal Enterprise Architecture" defines an EA as "a strategic information asset base, which defines the mission, the information necessary to perform the mission, the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs. Enterprise architecture includes baseline architecture, target architecture, and a sequencing plan" [13].

An EA is distinct from an EA framework, though the two are sometimes confused. A framework is a logical structure for classifying and organizing complex information. An Enterprise Architecture framework provides an organizing structure for the information contained in and describing an EA. The framework does not contain the EA itself. Many organizations can use the same EA framework, but each EA with its content is organization-specific.

Since Zachman introduced his framework [14] providing a deeper, more detailed understanding of the enterprise architecture, a number of other frameworks have been proposed. The list includes well-known frameworks, such as the Federal Enterprise Architecture Framework [13], the Open Group Architecture Framework TOGAF [2], the DoDAF: DoD Architectural Framework (DoDAF) [15], the ArchiMate Framework [16], and the Quasar Enterprise [17].

Much more than just a list of IT standards to be followed by an enterprise, EA must cover the entire information system management process in order to provide a common basis for understanding and communicating how systems are structured to meet strategic objectives.

B. Enterprise Architecture Layers

Although that no complete set of architecture description techniques, that fully enable and exploit integrated enterprise modeling, exist [18]-[21], in the current practice of organizations, enterprise architectures is modeled as a layered organization. The layers that are usually recognized in this context are the business layer, the application layer, the data layer and the technology (IT) layer (Fig. 4).

1) Enterprise architecture business layer

The Business Architecture view describes the "what" of the business model, activities, processes, functions, information and metrics for the organization. Building the business Architecture is the crucial step that is necessary to link business processes to the Strategic Intent of the organization.

Based on the Mission and the Vision of the organization, the core business processes of the organization can be identified. A process hierarchy or process decomposition can be established. This decomposes each process of the Process Architecture into its sub-processes, the lower level and therefore more specific sub-processes or process steps. This can be done for each process, until such a concrete and low level of abstraction is reached that further decomposition does not logically makes sense.

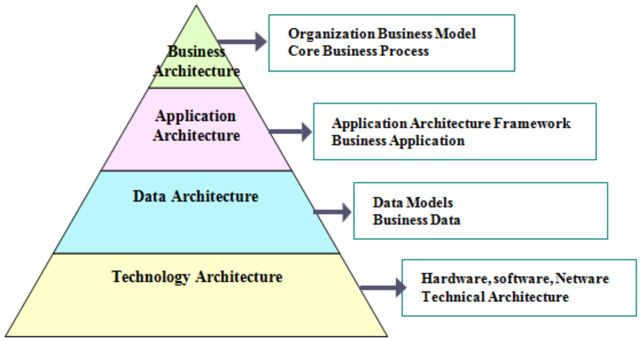

Fig 4. Enterprise architecture layers.

2) Enterprise architecture application layer

The Application Architecture view describes the "what" applications and "how" the organization uses Application software for automating Business processes. This layer encompasses the application systems to be deployed, their interactions, and their relationships to the core business processes of the organization.

The main structural concept for the application layer is the application component. This concept is used to model any structural entity in the application layer: not just software components that can be part of one or more applications, but also complete software applications, sub-applications or information systems, such as the ERP modules, the SCM system, etc. Each component can be associated to one or many process or sub-process belonging to the business layer hierarchy.

3) Enterprise architecture data layer

The Data Architecture view addressees the "what kind of data" and "how" question. It covers the persistent data used and modified by applications.

The Data Architecture is generally divided into two worlds: the Operational environment mainly constituted by business domains with "production databases", and the Decisional environment with "Datawarehouse", "Datamarts", Data manipulation and migration tools, Data Analysis and Restitution applications. Each data component is used by one or many application components to achieve one or many process goals.

4) Enterprise architecture technology layer

The technical architecture view addresses the "how" question. It describes the system software and hardware services necessary to run applications and store data for any type of business. Technology Architecture examines the underlying technologies that are required to run the organization. It defines the components or building blocks that make up an overall information system, and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

C. Relationships between EA Layers

While definition of each layer of an EA is an important first step in the EA development process, the true value of EA is found in the knowledge gained from the relationships among those layers.

Architecture links are defined to show the relationships and dependencies across and within architecture layers (Fig. 5).

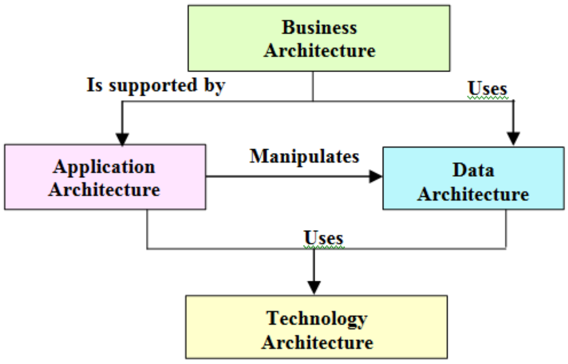

Fig. 5. Relationships between EA layers.

In this work, the definition of architecture links represents the second necessary step after defining all components of EA layers. The basic components (e.g. Database, Operating system) will be representing by a simple agents (DRA). The composite components (e.g. web service, procedure) will be representing by a composite agent (MARDS). The development of the structure of the principal composite agent, corresponding to the overall EA, will be based on the internal links between basic/composite components of each layer, and the external links between basic/composite components from different layers.

In the next section, we generalize the principle applied for specifying and modeling of organizational systems to EA.

IV. EA Business Modelling Based on MARDS: Example

In the [10], [11], we have used PAMS tool for developing modeling two business processes: the IT Project management (ITPM) process and the IT Strategic Plan (ITSP) process of an Information Systems Department (ISD) of a company.

So, we are going to follow the same steps for modeling an example of EA model. For thus, we should:

1) Specify the basic components (DRA agents) of each layer.
2) Identify and specify :
   - The first composite components (MARDS agent) of each layer if its exists;
   - The intermediate composite components from basic/composite components of different layers.
   - The principal composite component that represents the overall EA;
3) Implement all specified components in the PAMS tool which will generate automatically the Business process Model according to the selected initial action (e.g. Process call) of the principal composite component.

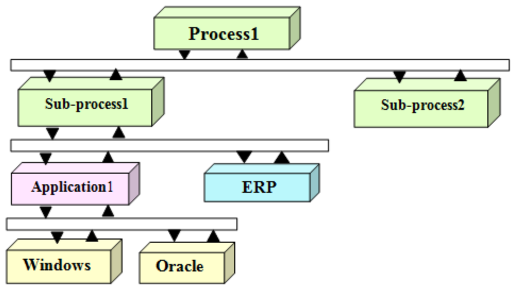

Fig. 6 (a). The MARDS structure of an EA model example.

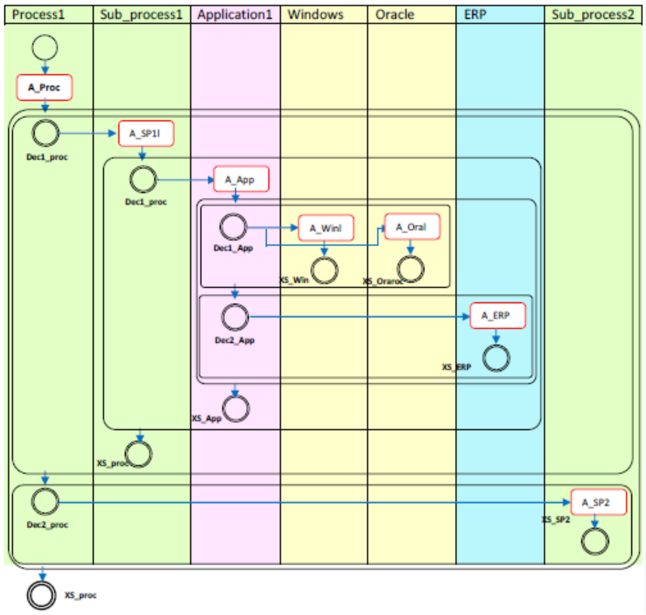

Fig. 6 (b). The business model of an EA model example.

A. The Structure of an EA Model Example

Fig. 6 (a) represents the MARDS structure of a simple example of an EA model. We are used the same colors chosen for EA layers to represents their basic/composite components.

In This EA model, the basic components are: Windows, Oracle and ERP; the intermediate composite components are: application1, sub-process1 and sub-process2; and the principal composite component is Process1

B. The Business Model of the EA Example

The Fig. 6 (b) displays the Business model of the EA model. The action A_proc received by Process1 component generates two decisions {Dec1_proc, Dec2_proc}. Each decision corresponds to a several sub-actions received by Sub_process1 component {Dec1_proc, A_SP1} and by Sub_process2 component {Dec2_proc, A_SP2}.

Every sub-action received by any composite component will be realized and modeled as a sub-process. For example, the sub-action A_App received by the Application1 component generates two decisions {Dec1_App, Dec2_App}. The first sub-decision Dec1_App generates the {A_Win, A_Ora} actions for {Windows, Oracle} basic components. The second sub-decision Dec2_App generates the {A_ERP} action for {ERP} basic component. The sequencing of the two sub-decisions corresponds to the sub-process of the A_App sub-action.

V. CONCLUSIONS AND FUTURE WORK

Enterprise Architecture is an invaluable tool for any organization looking to improve their ability to manage and implement changes to their business and IT systems. By applying a new approach of business modeling to EA, organizations are able to gain a better understanding of the elements that make up their enterprise - business, application, data and technology layers.

In this paper, we have presented a simple EA model for explaining the concepts of this approach. To illustrate its advantages, others examples of EA will be developed and modeled in the future work. We can also exploit the functionalities of PAMS tool by generating the BPEL processes of the EA business models. So, a new service oriented modeling approach will be proposed.

REFERENCES

[1] J. Zachman, "A framework for information architecture," *IBM Systems Journal*, 1987, vol. 38, pp. 2-3.

[2] The Open Group, "The open group architecture framework (TOGAF)," 2002.

[3] R. Winter and R. Fischer, "Essential layers, artifacts, and dependencies of enterprise architecture," *Journal of Enterprise Architecture*, May 2007.

[4] T. Do, M. Kolp, and A. Pirotte, "Social patterns for designing multiagent systems," in *Proc. the 15th International Conference on Software Engineering and Knowledge Engineering (SEKE 2003)*, 2003, San Francisco, USA.

[5] Bounabat, "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.

[6] A Aaroud, S. E. Labhalla, and B. Bounabat, "Modelling the handover function of global system for mobile communication," *The International Journal of Modelling and Simulation*, ACTA Press, vol 25, no. 2, 2005.

[7] A. Aaroud, S. E. Labhalla, and B. Bounabat, "Design of GSM handover using MARDS model," in *Proc. IEEE Conf. International Conference for Information Technologies and Application (ICITA)*, 2004, Harbin, China.

[8] M. Berrada, B. Bounabat, and M. Harti, "Qualitative verification of Multi-Agent reactif decisionnal systems using business process management notation," *The IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06)*, 2006, Hong Kong.

[9] M. Berrada, B. Bounabat, and M. Harti, "Process management oriented modeling and simulation for Multi-Agents reactive decisional systems," *The IEEE Information and Communication Technologies International Symposium (ICTIS'07)*, 2007, Fez, Morocco.

[10] M. Berrada, B. Bounabat, and M. Harti, "Modeling and simulation of Multi-Agent reactif decisionnal systems using business process management concepts," *International Review on Computers and Software (IRECOS)*, vol. 2, no. 2, pp. 159-169, March 2007.

[11] M. Berrada, B. Bounabat, and M. Harti, "Process management oriented qualitative verification for multi-agents reactive decisional systems," *InfoComp-Journal of Computer Sciences*, vol. 6, no. 4, pp. 27-35, December 2007.

[12] BPMI.org. (2006). Business Process Modeling Notation Specification. *OMG Final Adopted Specification*. [Online]. Available: http://www.bpmn.org/

[13] Federal CIO Council, "A Practical Guide to Federal Enterprise Architecture," 2001.

[14] J. F Sowa and J. A. Zachman, "Extending and formalizing the framework for information systems architecture," *IBM Systems Journal*, 1992, vol. 31, no. 3, pp. 590-616.

[15] Department of Defense Architecture Framework Working Group, "DoD architecture framework," vol. I-II, Deskbook, 2003.

[16] M. M. Lankhorst *et al*., *Enterprise architecture at work, modeling, communication and analysis*, 2005, New York: Springer.

[17] E. Gregor, H. Andreas, H. Bernhard, J. Oliver, L. Marc, J.-P. Richter, V. Markus, and W. Johannes, *Quasar enterprise: Anwendungslandschaften serviceorientiert gestalten*, 1st ed., Heidelberg: dpunkt.verlag, 2008.

[18] H. Jonkers, "Towards a language for coherent enterprise architecture description," in *Proc. 7th IEEE International EDOC'2003*, in Brisbane, Australia.

[19] P. Sousa, A. Caetano, A. Vasconcelos, C. Pereira, and J. Tribolet, "Enterprise architecture modeling with the unified modeling language," *Enterprise Modeling and Computing with UML*, IRM Press, November 2005.

[20] M. Assmann, "Model-based evaluation of service-oriented enterprise architectures," PhD Thesis, University of Paderborn, 2009.

[21] M. E. Iacob, H. Franken, and H. van de Berg, *Enterprise Architecture Handbook – method, language and tools*, BiZZdesign Academy Publishers, 2007.

Mohammed Berrada was born on April 19, 1977 from Fez and got PhD in Computer Sciences from the faculty of Sciences, Sidi Mohammed ben Abdellah University, Fez, Morocco, in 2008.

Assistant Professor in ENSAF (National School of Applied Sciences of Fez), Sidi Mohamed ben Abdellah University, Fez, Morocco. The major fields are Multi-Agent systems, Enterprise Architecture, Modeling, Web services, Autonomic computing.

Bouchaïb Bounabat was born on March 8, 1966 from Marrakech and got PhD in Computer Sciences, Marrakech University, Morocco, in 2000. University Doctorate in Computer Sciences/Engineering Sciences, Telecom National Institute, Evry University, Paris, France, in 1993.

Professor in ENSIAS (National Higher School for Computer Science and System analysis), Rabat, Morocco. The major fields are Enterprise Architecture, Information System Governance, Reactive systems.