An agent oriented information system: an MDA based development

Abdelaziz Elfazziki [1], Sana Nouzri [2] and Mohamed Sadgal[3]

1 Computer system engineering laboratory, Faculty of Sciences Cadi Ayyad, university Morocco, B.P. 2390, Marrakech

2 Computer system engineering laboratory, Faculty of Sciences Cadi Ayyad, university Morocco, B.P. 2390, Marrakech

3 Computer system engineering laboratory, Faculty of Sciences Cadi Ayyad, university Morocco, B.P. 2390, Marrakech

Abstract

Information systems (IS) development should not only accomplish functional models but also conceptual models to represent the organizational environment in which it will have to evolve and must be aligned with strategic objectives. Generally, a significant innovations in the enterprise, is to organize its IS around its business processes. Otherwise, business models must be enriched by the agent paradigm to reduce the complexity involved in solving a problem by the structuring of knowledge on a set of intelligent agents, the association between agents and activities and collaboration among agents. To do this, we propose an agent oriented approach based on the model-driven-architecture (MDA) for the information system development. This approach uses in its different phases, the BPMN language for the business processes modeling, AML language for the agent modeling, and JADEX platform for the implementation. The IS development is realized by different automated mappings from source models to target models.

Keywords: *Meta-modeling, information systems development, business processes, agent technology, BPMN language, Model Driven Architecture. JADEX platform, AML language.*

1. Introduction

The information systems (IS) complexity does not stop growing and their development becomes increasingly complex, expensive and difficult. The use of new advanced systems engineering techniques is necessary to help controlling the complexity of the encountered problems and to answer required flexibility and adaptability [10]. The specification of a particular IS requires some agility where the developer can define semantics which most fit his system. Other ways, to guarantee a development quality, the environment of development should allow expression and validation of properties during the development process.

In model-driven development, IS developers must face the challenge of building IS solutions that are aligned with business processes in order to satisfy business requirements. To this endeavor, they must understand those business processes, which require extensive collaboration with domain experts and analysts. Because business processes are typically expressed using dedicated modeling notations (BPMN). Thus, IS developers are required to master these particularities. In order to support these requirements in performing the business information system alignment, we define a mapping from domain model to BPMN models and from BPMN models to agent models and from agent models to a specific platform model. This approach allows the re-expressing business processes in a language that is closer to the IS developers.

To deal with IS agility, conceptual modeling must not only represent the organizational environments of the IS, but also take into account the strategic objectives and business processes to understand the requirements of their development.

There are several modeling approaches available and each one of them has its own rules. Examples of other notions are Event-driven Process Chain (EPC) [8], XML Process Definition Language (XPDL) [9], UML Activity Diagram and Business Process Modeling Notation (BPMN) [1].

In this paper, we describe a mapping from business process diagrams to agent concepts that simplify the transfer of knowledge between the roles involved in the software development process. The approach adopted is based on Meta modeling, model driven architecture (MDA) [4], Multi-agent architecture derived from the enterprise structure. To realize this approach, we propose the use of an open source environment: BPMN [1] for the business processes modeling, AML [2] language for the agent modeling.

The objective of this work is to describe an agent oriented Model-Driven approach. This approach is therefore put

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

forward and based on a set of business models under continuous maintenance of business actors to reveal the current business needs, models have being associated with adaptive agents that interpret the captured requirements to behave dynamically, always fulfilling current requirements. Consequently, the maintenance of the models is the maintenance of the information system. This provides a means of model-based adaptation rather than code-based adaptation.

The reminder of this chapter is structured as follows. Section 2 provides an overview of the BPMN language. The MDA concepts descriptions are presented in third section. The section 4 describes the business processes meta-modeling, where we describe a BPMN meta-model and a multi agent system meta-model. The section 5 describes the proposed approach. Section 6 describes the analysis domain. Section 7 describes the business process agent modeling, where we illustrates the mapping between BPMN concepts and BDI concepts. Section 8 provides an overview of Agents interactions. In the Section 9 we proceed to the validation of the approach by a case study. Section 10 presents a synthesizing of models and their dependencies. Finally, section 11 concludes this paper.

2. Business Process Model and Notation (BPMN)

Major headings are to be column centered in a bold font without underline. They need be numbered. "2. Headings and Footnotes" at the top of this paragraph is a major heading.

The main concepts in Business Process Modeling Notation (BPMN) are Flow Objects that are connected via Sequence and Message Flows. They structure activities into events and tasks for decomposing and joining Sequence Flows. They describe the sequence in which the several Flow Objects have to be accomplished, while Message Flows describe the exchange of messages. Thus, BPMN combines the definition of local workflows and the interaction between them. It is adopted by the Object Management Group; it is a graphical notation to describe business processes. Its objective is to provide a notation understandable by business users and to create a standardized gateway to bridge the gap between modeling business process and languages such as BPEL4WS. The BPMN 2.0 [7] offers the following diagrams:

- Conversation Diagrams: provide an overview of communications between actors.
- Choreography Diagrams: focus on the details of the conversation between several actors, and are often linked to specific nodes of conversation.
- Collaboration diagrams: focus on the messages exchanged between actors.
- Process diagram: focus on the flow sequence in a single process within an actor.

The BPMN is a standard for modeling business processes; it provides a graphical notation for specifying business process in a business process diagram (BPD) [5] based on a flowcharting technique very like the UML activity diagram (UML) [6].

3. Model driven architecture (MDA)

The MDA is a software development approach, proposed and supported by Object Management Group (OMG). Its basic principle is the elaboration of platforms Independent Models (PIM) and the transformation of these PIM into Platforms Specific Models (PSM) for the systems concrete realization. The used techniques are therefore in most cases techniques of modeling and techniques of models mapping.

MDA is a systematic use of models as firstly engineering artifacts throughout the engineering development process [4], paradigm introduced by the Object Management Group (OMG).

MDA is a pertinent emerging paradigm that it focuses for portability, interoperability and reusability, three non-functional criteria that are very important for agile systems design. MDA defines three models:

- A computational independent model (CIM) is a view of a system that does not take into account the system structure details. It uses a vocabulary that is familiar to the domain actors in question as it is used for system specification.
- A platform independent model (PIM) is a view of a system that on one hand provides a specific technical specification of the system, but on the other hand defines a specified degree of platform independence so as to be suitable for use with a number of different platforms.
- A platform specific model (PSM) is a view of a system combining the specifications in the PIM with the details that specify how that system uses a particular kind of platform. In this work the CIM, PIM and PSM are the models outputted by the domain analysis, design and implementation phases respectively. Each of these models is produced by applying simple mapping rules to the previous phase model and this mapping is noticeable, that is it can be used in reverse engineering.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

4. The Meta-modeling

In this approach, we propose the use of a series of meta-models with their transformation rules from the starting analysis meta-model (the domain meta-model) until the implementation meta-model. Firstly, we adopt the analysis meta-modeling, and then the business meta-modeling based on the BPMN concepts to graphically represent business processes and the agent meta-model in order to describe a dynamic information system enriched by the agent paradigm. It is not necessary to rely on a unified framework for business processes meta-modeling and agent paradigm, but just rely on the problem statements to design these two meta-models.

4.1 The BPMN meta-model

In the BPMN, a process is divided into one or more (Pools) corresponding to the participants. A (pool) can be divided into several lanes (Lane). It represents the involved actors, organizational roles, or departments, each containing part of the process associated with the actor in the same domain of control. The process basic units are tasks, sub processes, and events interconnect by sequence flows within a pool and shows message flows exchanged between pools. BPMN offers a Gateway, it can be used for branching, splits, conditional splits, merges, or synchronizing joins.

The objective in this stage is to be based on the BPMN meta-model concepts, to define the transformations rules guiding at the PIM level and to instantiate the CIM models at modeling level. The figure 1 describes the BPMN meta-model.

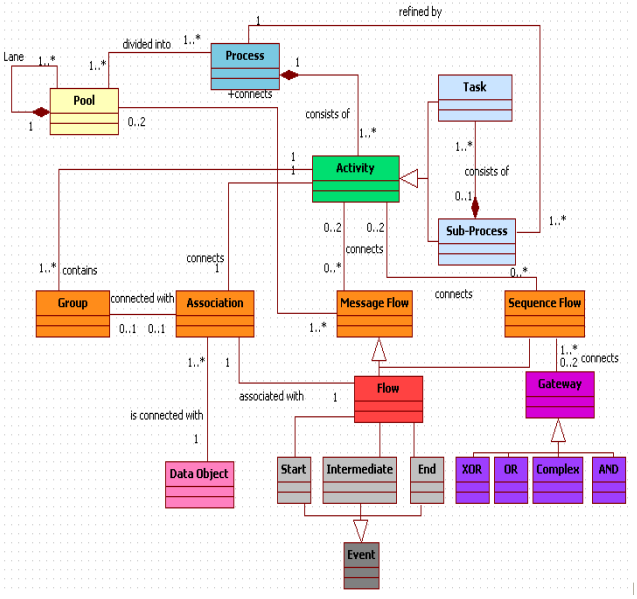

Fig. 1 BPMN Meta-model.

4.2 The agent BDI model

In literature, there is no single, universally accepted definition of the term agent. Different definitions, however, certain properties are required to translate a complete description of agent. In our opinion, the agents following the BDI (belief, desire and intention) [3] are best adapted to capture all functionality that is expressed with BPMN. Thus, we propose to use agents that are able to perform BDI reasoning related to the objectives, plans, intentions and beliefs.

BDI architecture is designed by starting of the model "Belief-Desire-Intention", rationality of an intelligent agent. In the following, we present an informal, intuitive meaning of these three elements in a model BDI. The model has an associated formal logical theory, but we will not go into details of it.

The B = Belief : The agent beliefs are the information that the agent has on its environment and on the other agents of the system. Beliefs may be incorrect, incomplete or uncertain, and for this reason, they are different from the agent's knowledge: information is always true. Beliefs can change over time, its ability to perception or interaction with other agents collects more information.

The D = Desire : The desires of an agent represent the environment states, that the agent would like to see done. An agent may have contradictory desires, in which case, it must choose between its desires, a subset that is consistent. This subset consists of its desires that are sometimes identified with the agent goals.

The I = Intention : The intentions of an agent are the desires that the agent decided to do or the actions which it decide to do to accomplish its desires.

5. The proposed approach

To answer the requirements of the information systems development, we propose the adoption of a business processes oriented approach. This approach is based on concepts of Meta modeling, agent technology and model driven architecture. It uses a multi-agent systems architecture mapped from the considered enterprise structure. And it specifies the life cycle of the system as process of production by using iterative refinement and incorporation of models. It is centered on elaborating and transformation of models.

5.1 The architecture

This approach is built around the MDA architecture with four modeling levels. The first one constitutes the meta-

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

meta model; it serves for defining the syntax and semantics according to which meta-models are going to be applied. The second one represents the Meta model layer, in our context, it seems necessary to be able to represent in an independent way concepts of the application domain and to specify their properties. Then these concepts will be correctly structured into agents. This separation of aspects is illustrated by the separation of layers: domain layer (business domain), the agent oriented layer and the implementation platform layer.

5.2 Development approach

Our main objective in this chapter is to propose an original approach for agile information system development. This approach takes into account the diversity of agent-oriented concepts based on the meta-modeling of multi-agent systems. It also considers the dynamic of information systems based on meta-modeling of business processes. It constitutes a consistent approach that preserves the semantics of concepts throughout the development cycle while working on the principle of model-driven architecture. The aim of relying on an architecture is to separate the different views of the system (organizational view, business view, technical view, etc.). It is specifically based on the MDA to express how to transform, or match the concepts of the various models in order to have a global representation of multi-agent system. The MDA paradigm can also cover the different phases of the development cycle by adopting the transformation rules between models. This approach ensures consistency of the system during the different phases of the development process. The applied development approach is based on the following steps:

- Definition of the development framework formed by the meta-models and transformation rules,
- Use of the development framework by creating models based on meta-models defined and applying transformation rules between these models.

This approach is more coherent for integrating the various aspects of multi-agent system and facilitates the work of developers throughout the development process. In this approach, we propose to follow the order of meta-models hierarchy using while starting with the business meta-model until the specific modeling step and transiting from the described stages: as shown in figure 2:

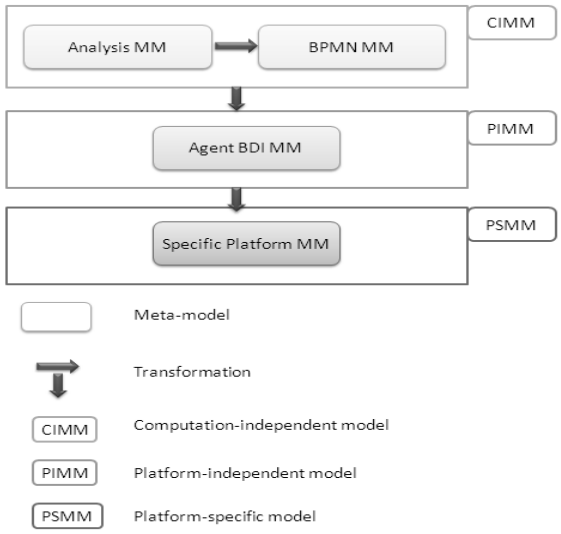

Fig. 2 The meta-models hierarchy

6. The domain analysis

In the analysis phase, the analyst needs to start capturing the functionality behind the system under development. In order to do this, he needs to start thinking not in term of goal but in term of what will the system need to do and who are the involved actors in each activity. Thus, the use case diagram helps to visualize the system including its interactions with external entities.

Before starting the BPMN modeling, we must first extract the application process, it is necessary to:

- Identify domains of enterprise,
- Associate with each domain a sub-information system, which presents a part of global information system,
- Identify the actors in each domain,
- Identify the functions associated with the actors,
- Define use cases from each sub Information System,
- Associate with each information system a process.

The figure 3 shows the decomposition of the global enterprise information system to several sub- information systems according to domains to identify use cases and then the processes, and subsystems.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

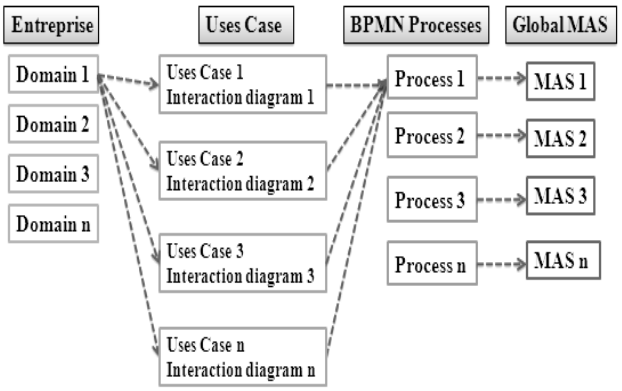

Fig. 3 Relationship between information system and process

7. Business Process Agent Modeling

We chose to represent a BP by MAS to give place to an agent modeling of a BP. Thus, a business process will be represented by MAS where the concept of actor is represented by an agent (based on autonomy); the activities are modeled only in terms of goal and plan and are under the responsibility of an agent which the internal behavior is transparent. If multiple agents are in charge of an activity, like protocols social convention represent a standard framework of this collaboration. The agent modeling that we propose recommends a classification of agents into two categories: Intelligent Agents and reactive agents. The modeling approach that we propose is done in two steps:

- The first concerns the collection of requirements and domain analysis realized by using UML2 which consists of defining functional requirements, non functional requirements and the identification of principal artifacts that will form the solution. Concerning these artifacts, it is a first estimate of the classes that realize the application. They are divided into three categories: interface class (or boundary), the controller class and entity class (BEC Class Diagram). Each use case must be made by the collaboration of these three kinds of classes.
- The interface classes: they represent the interface between the actor and the system.
- The control classes: they represent the process activities.
- The business classes (or entity classes): These are the data invoked by the activities. They correspond to persistent information: they are stored on permanent support such as a database. After formalizing the requirements of all use cases, relationships, the actors involved, the scenarios of each use case, interaction diagrams and the BEC class diagram are defined.
- The second step is the analysis stage, it concerns the activities definition, sequences, control flow and complete the elements of BEC class diagram. This step will be achieved by the use of the BPMN language.
- To develop multi-agent model, we propose two stages. The first one uses a mapping from BEC class diagram and interactions diagram to the agent concepts. The second one uses a mapping from BPMN description to agent concepts. The figure 4 below illustrates this proposal.

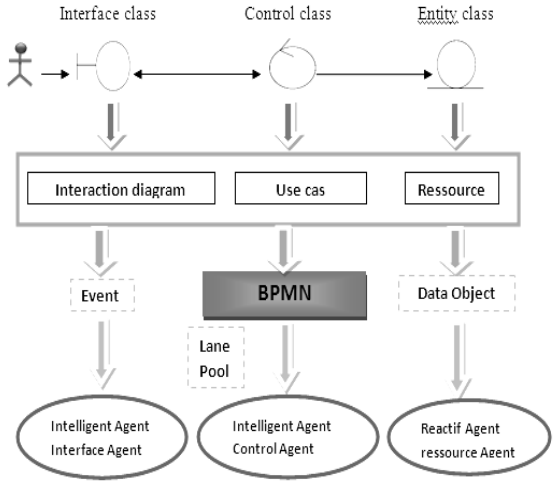

Fig. 4 Different level of mapping

7.1 Mapping from BPMN to BDI agent

The business process modeling defines a global view of the system. It can be used as a basis for the development of agent behaviors. This by identifying for each agent: its plans, its goals, its interactions, its states and its resources. Normally, the agent derivation behaviors from a BPN diagram is highly dependent on the typology of the agents and how their behaviors are implemented in a given environment; this derivation comes essentially from a set of rules based on the meta-model agent concepts and the BPMN meta-model. Generally there are two types of pools: simple pools and structured pools into lanes. From each simple pool an agent behavior is derived and from each lane of a structured pool is an agent behavior is derived. These behaviors cover all activities sequences of an agent. In addition, use cases, interaction diagrams and BEC diagrams are used to identify agents with their types and their interactions. Figure 2 presents an overview of the

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

models with their matches. The main transformation rules are presented in the following table:

Table 1: Mapping from BPMN concepts to BDI concepts

BPMN concepts	*Agent BDI concepts*
Process	One or more pools
Pool	Group of agent
Lane	Agent
Sub process	Plan
Task	Action
Start event	Trigger plan
Intermediate event	Message or precondition to execute of a plan
End event	Goal
Data object	belief

8. Agent interactions

The inter-agent interactions are three types:

- **Interaction between interface agents and control agents**: these interactions are deduced from the interaction diagrams for each external actor.
- **Interactions between control agents**: these interactions are deduced from the relationships between use cases and transversal relations between processes.
- **Interactions between control agents and business agents**: these interactions are deduced from the association between the control class and the entity classes or directly from the BEC diagram. The figure 5 illustrates these interactions.

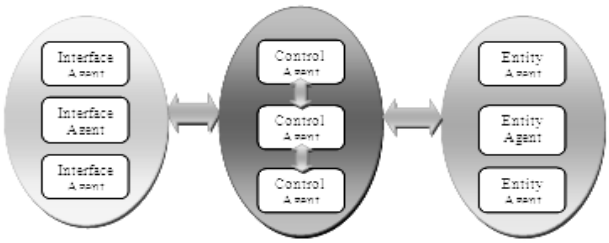

Fig. 5 Multi-Agent Model of a business process

9. A Case Study

To illustrate our proposal, we propose the description of an appointment management for consulting a doctor.

9.1 The domain Modeling

Uses case diagram
The use case diagram of the appointment management for consulting a doctor is described in the figure 6.

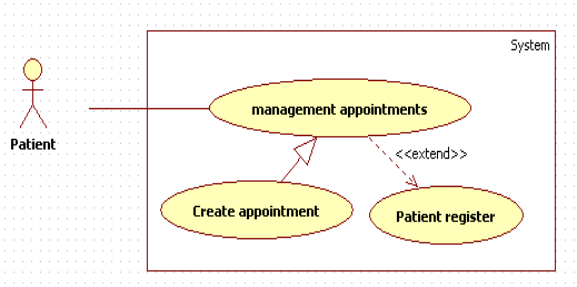

Fig. 6 Use case Diagram

The BEC diagram
The BEC class diagram of the appointment management for consulting a doctor is described in the figure 7.

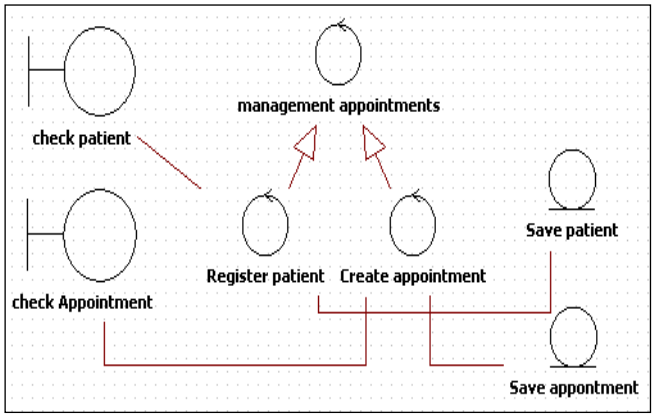

Fig. 7 The BEC class diagram

9.2. The BPMN model

The BPMN diagram of the appointment management for consulting a doctor is shown in the figure 8.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

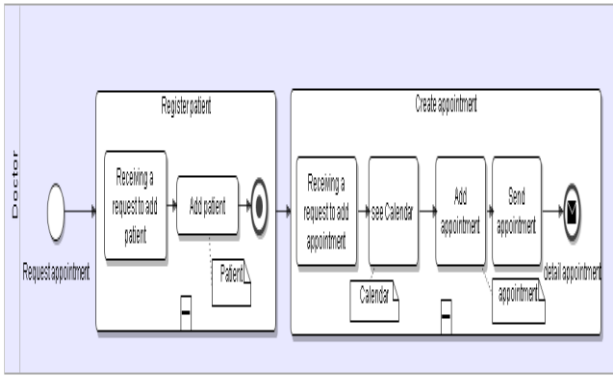

Fig. 8 The BPMN model

9.3 The agent modeling

For the development of agent-based model, it will be done in two stages. The first uses a mapping from the BEC class diagram to the agent concepts. The second will be done by using a second mapping from BPMN description to the agents' concepts. The figure below illustrates the first mapping.

By applying the first mapping, we identify with the BEC analysis classes three types of agents:

- Interfaces agents: are intelligent agents that are getting external events and activate the control classes.
- Controls agents: are intelligent agents engaged to perform internal operations system (control classes).
- Entities Agents: are intelligent agents dealing with business classes of system, which are used by the control classes.

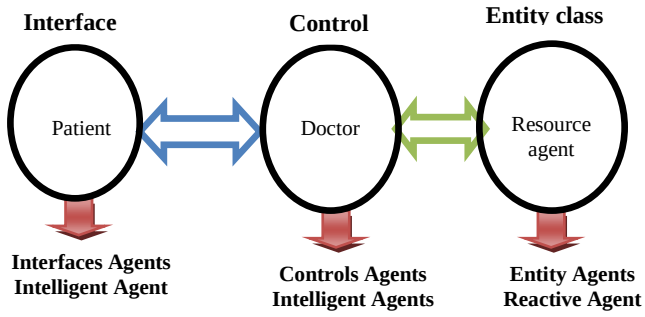

Fig. 9 Agent typology

9.4 The agents' behaviors

The behavior of an agent is defined by specifying its plans and actions. Agents have a set of plans that are selected for execution according to their goals and their states. In this paper, we propose to deduce goals and plans of an agent from the BPMN diagram by using transformation rules. We consider a plan as an activity. The two concepts, plans and activities consist of actions and define the sequence of the actions. The figure below illustrates the second mapping.

By applying the second mapping, we define the behavior of that will be described by the concepts of BDI agent. The following table defines the behavior instances of each agent:

Table 2. Agent behaviors

BDI Agents	Behaviors			
	Goals	Plans	Actions	Beleinfs
interface Agent	Validate data patient	check patient	Display registration Form	List days Férier
			Validate registration Form	
			Send request to add patient	
	Validate request appointm	Verify appointment	Display appointment form	
			Validate request appointment	
			Send request appointment	
			Display appointment	
Control Agent	Manage appointment	Create appointment	Recieve request appointment	Calendrier
			Consult calendar	
			Add appointment	
			Send appointment	
	Manage patients	Register patient	Receive request add patient	
			Add patient	
Entity Agent	Manage datas	Save patient	Recieve data patient	
			Store patient	
		Save appointment	Recieve data appointment	
			Store appointment	

10. An overview of models and their dependencies

In the figure below, we present a diagram summarizing the different kinds of the approach models (the domain model, the activities model and agent model activities) and their dependencies.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

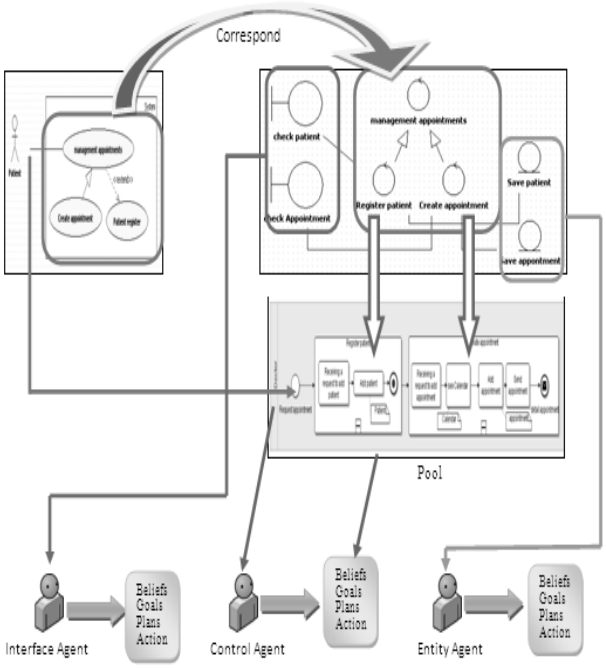

Fig. 10 models and their dependencies

11. Conclusion

This paper describes a model-driven approach to close the existing gap between business requirements specified using BPMN and agent technologies. It presents a multi-agent approach based on the MDA paradigm and using a business process meta-modeling for the information system development. Thus it uses several meta-models and models to transform conceptual representations of an organization into an operational information system. It describes an approach based on the mapping process from BPMN models to agent models.

The approach would allow developer involved in the implementation to gain many benefits such as reducing the communication time or seamlessly synchronized BPMN model with the implementation. It is also aimed by Business Processes allowing to optimize them and to ameliorate their agility.

However, experiment with real case study is necessary to validate the suggested approach. The question that has to be answered is how much information has to be manually added on the PIM level after applying BPMN to AML model transformation to generate executable agent code.

Acknowledgments

The authors wish to acknowledge the contributions of other members of the engineering laboratory of computer systems for their helpful discussions and the availability of all resources that have helped make this work in the best conditions. They would also like to thank the anonymous reviewers for their remarks and suggestions.

References

[1] Stephen, A. (2006)." Introduction to BPMN", IBM, Retrieved from http://www.bpmn.org/Documents/OMG_BPMN_Tutorial.pdf

[2] Cervenka, R et al. (2004). "AML: Agent Modeling Language Toward Industry-Grade Agent-Based Modeling", Proceedings of AOSE: agent-oriented software engineering V, ISBN: 3-540-24286-4, New York NY, 19 July 2004.

[3] Holger Endert, Tobias K¨uster, Benjamin Hirsch, and Sahin Albayrak. (2007)." Mapping BPMN to Agents: An Analysis", In proceeding Worklshop MALLOW-AWESOME'007(Agent, Web Services, and anthologies Integreted methodologies. Retrieved from < http://awesome007.disi.unige.it/EndertEtAl-AWESOME007.pdf>

[4] Miller, J. & Mukerji, J. (2003). "MDA Guide Version 1.0.1", OMG, Retrieved from <http://www.omg.org/cgi-bin/doc?omg/03-06-01>

[5] Richard, C. (2004). "An XML Representation for Crew Procedures", In: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050202022_2005202152.pdf, Available from: <hdl.handle.net>

[6] Stephen, A. (2004). : "Process modeling notations and workflow pattern", In Workflow Patterns, pp. (265-294), Future Strategies Inc, Retrieved from < http://www.bpmn.org/Documents/Notations_and_Workflow_Patterns.pdf >

[7] OMG, (2011) Document Number: formal/2011-01-03 Standard document URL: http://www.omg.org/spec/BPMN/2.0

[8] Davis, R and BRABÄNDER, E. (2007). "EPC Event-driven Process Chain". Aris design platform. Getting Started with BPM, in: http://www.cefetes.br/pse/pos_producao/disciplinas_new/gp/disciplina/gerenciamento_de_operacoes_e_processos_040_EPC.pdf

[9] Workflow Management Coalition (WFMC), (2001). Workflow Process Definition Interface -- "XML Process Definition Language", Number WFMC-TC-1025, Status – Draft 0.03a (Alpha Status). Retrieved from < http://xml.coverpages.org/XPDL20010522.pdf

[10] Salah Eldin Abdelrahman1, and Mohammed Badawy, "HASoC for Developing a Software System", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011, ISSN (Online): 1694-0814

Aziz El Fazziki is Professor of Computer Science at Marrakech University, where he has been since 1985.He received an M.S.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

from the University of Nancy (France) in 1985. He received his Ph.D. in Computer Science from the University of Marrakech in 2002. His research interests are in soft ware engineering, focusing on information system development. In the MDA arena, he has worked on agent based systems, service oriented systems an decisional systems. Professor Aziz El Fazziki is co-author agent based image processing, and is the author of over fifty papers on soft ware engineering.

Sana.Nouzri is a member in the research laboratory (computer system engineering) of the Department of Computer Science at the Faculty of Semlalia (Marrakech, Morocco) for the preparation of her PhD; she worked on agent-oriented approach directed by models for the development of information systems. Her current research interests include artificial intelligence, agent-oriented methodologies, the model driven architecture, transformations and development agent-oriented languages.

Mohamed Sadgal is Professor of Computer Science at Marrakech University. . His research interests are in artificial intelligence.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.