Towards an MDA Based Multi-agent Approach for Information System Development

Abdelaziz EL Fazziki, Sana Nouzri, Mohamed Sadgal
Département Informatique, Faculté des Sciences SEMLALIA
Université Cadi Ayyad, Morocco

Abstract

To tackle the problem of Information Systems (IS) evolution, conceptual modeling must not only represent the organizational environments of the IS, but also take into account the strategic objectives to understand the requirements of their development. In this paper we propose a business process oriented approach. This approach is based on Meta modeling, model driven architecture, Multi-agent architecture derived from the enterprise structure. To realize this approach, we propose the use of the Visual Agent as a software development environment which allows the necessary assistance from the specification phase until the implementation phase.

1. Introduction

The complexity of the information systems (IS) 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 to control the complexity of the encountered problems and to answer required flexibility and adaptability.

The specification of a particular system requires some flexibility where the developer can define semantics which most fit his application domain. On the other hand, to guarantee a development quality, the environment of development should allow expression and validation of properties during the development phase.

Effectively, the information systems are complex systems requiring a great number of interconnected components. In such context, it is necessary to be sure that the not functional and functional properties remains kept during the development phase. The application of a modeling approach changes from one project to another, therefore from one situation to another; consequently we prove the need to have adaptive components, which is the case of multi agents systems.

The main objective of this work is to contribute to improving the IS engineering. With this intention, we propose an IS development phase based on Meta modeling, model driven architecture and multi-agent systems. It also proposes a meta-modeling business processes with their enhancement with the agents paradigm to make them more adaptive. That would allow enterprises to have two fundamental properties: flexibility in the design and execution of their business processes and the ability to federate existing heterogeneous applications. It allows dynamically managing processes, adapting them to the organizational level, easy access to heterogeneous data and responding to alerts on the running processes.

The remainder of this paper is structured as follow: In the second section we describe the model driven architecture. The third section is dedicated to the approach description. In the fourth section we review the different BP modeling approaches presented in the literature and describe the business processes modeling, where we offer meta-model synthesizing classical representations. The section five present the MDA principles and multi-agent systems. The section sixe shows the enhancement of the business processes (BP) by multi-agent system. The section seven presents the visual agent environment. Finally, we finish this work by a conclusion where we present our contributions and our perspectives.

2. Model Driven Architecture (MDA)

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

The MDA phase of development can be similar to a classical development phase. We effectively find there the same stages of development such as the requirements expression, the analysis, the comprehension, the realization, the validation and deployment.

The first produced model called Computer Independent Model (CIM) corresponds to the description in natural language of the application domain. It serves as base in the definition of the PIM model (Platform Independent Model) .The PIM model in a third stage is refined to define the specific aspects

of the development paradigm which is going to be used in our case the Agent paradigm.

The fourth stage consists in producing, by applying an automatic transformation, a PSM model which is a platform specific model. After refinement of this kind of models, they serve for generating automatically a party of the final code which must be later refined and tested. The figure 1 shows the development phase [10], [14].

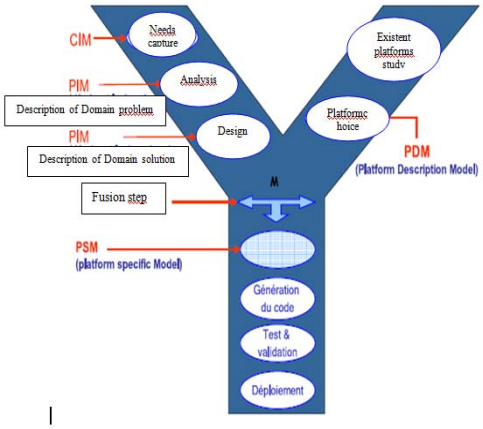

Figure 1. Life development cycle of MDA

3. The proposed approach

To answer the requirements of the information system's development, we propose the adoption of a business processes oriented approach. This approach is based on concepts of Meta modeling and model driven architecture. It uses a multi-agent systems architecture mapped from the structure of the considered enterprise. 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.

3.1 The architecture

This approach is built around architecture with four levels (MDA architecture). The figure 2 illustrates this architecture [10].

M3 constitutes the meta-meta model; it serves for defining the syntax and semantics according to which meta-models are going to be applied.

M2 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 agentified correctly. This separation of aspects is illustrated by the separation of layers: domain layer, the agent oriented layer and the implementation platform layer [10][18]: Therefore we identify three kinds of Meta models:

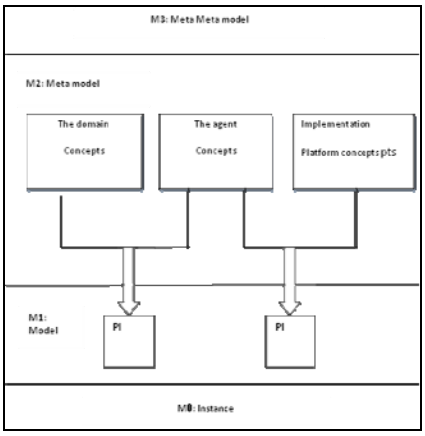

Figure 2. The four levels architecture

The M1 layer represents PIM and PSM. These two models are further acquired to the weaving of aspects belonging to the Meta model which we defined before. It is accomplished thanks to transformation rules. The Meta models domain: it represents concepts used in the application domain.

- The implementation Meta models layer: it represents concepts which are implemented by the platform.

- PIM is acquired by weaving the domain concepts and the agent concepts. We obtain in this model, an agentified description of the application domain. Another transformation is necessary which links agent oriented concepts in platform concepts. The application of these transformation rules generate PSM constituted by the application source code. The M0 layer represents the system layer. Its deployment from the source code generated in PSM is performed at the level of this layer.

3.2 Development phases

The proposed approach allows the developer to create his development framework which is specific to his application. The development framework is going to contain:

- Concepts and specific properties of the application domain,

- Concepts and specific properties of the multi agent system;

- The transformation rules between domain concepts and agent concepts;

- Concepts and properties on which are based the implementation platforms;

- The transformation rules between agent concepts and implementation platform concepts. The

application of the approach step is situated at two levels:

- The first level consists in defining the development framework formed by the Meta model and the transformation rules;
- The second level consists in using the development framework by creating models based on the defined Meta model and by applying transformation rules among them. For this, we propose the following development phase [14] (cf. Figure 3).

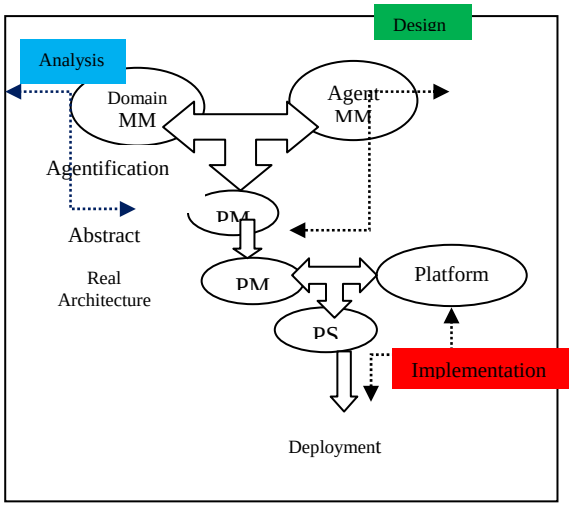

Figure 3. Development phase

3.3. Requirements relating to carry out the approach

To carry out this approach, it is necessary to have "tools" which will help particularly in the building of the development framework. These tools will be regrouped in a development environment. To provide these different elements, we adopt an open source environment of software engineering based on STARUML [11], MAS-ML [17] and ASF framework [18] and the Visual Agent [17] platform. The figure 4 shows the different mappings among the different frameworks of development.

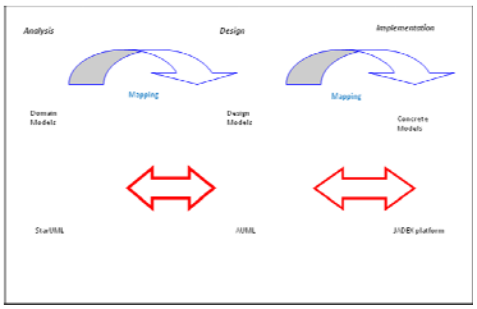

Figure 4. Frameworks of development

3.4. The approach steps

In our approach we propose to follow the order of meta-models using while starting with the functional meta-model until the code generation step and transiting from the described stages on the figure 5:

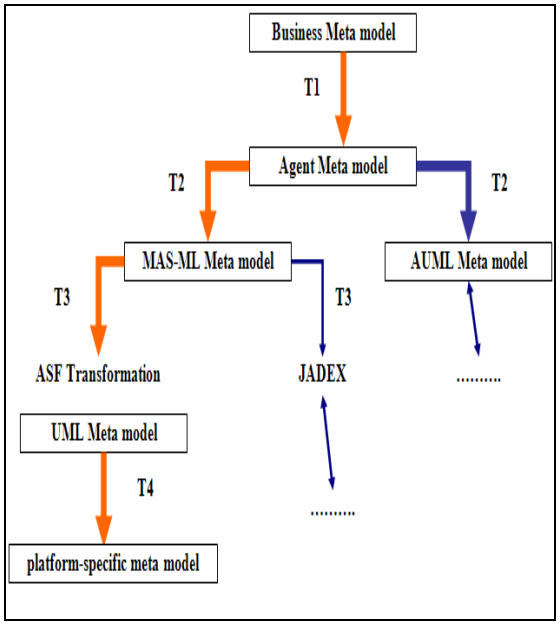

Figure 5. Hierarchy of the proposed meta-models

3.5. Transformations

The transformations from one meta-model to another that we propose are transformations by annotation or of marking. To arrive at the meta-model target, we use the mapping starting from the initial meta-model. Each model must be described by a meta-model, which defines its principal characteristics.

The MDA also defines a whole of consecutive transformations that should be applied to models in order to allow the transformation of the high level models abstraction into generated code.

In our approach, to define the business meta-model, we propose, to use a business processes meta-model, and in order to enrich this meta-model, we propose the use of MAS meta-model. To define PIMs, we adopt MAS-ML language [17] for representing MAS, since it is independent from the platform and in order to define PSMs, we propose to transform the MAS-ML models into UML models (OMG, 2005) by using the object Framework ASF [ ], the standard XML Meta-model Interchange (XMI). Finally the transformation from PSMs into generated code is carried out by the UML implementation tool STAR-UML (reference).

4. The approach Meta models

In our proposal, in order to build a multi-agent system we use several meta-models. At first, we start with the requirements meta-model which describes the system field, thereafter to ameliorate this meta-model, we propose a multiagent meta-model. Finally we use several mapping to get a generated code by using MAS-ML meta-model, UML meta-model UML and a specific platform meta-model.

4.1. Business process (BP) Meta-modeling

The BP modeling of an enterprise is to represent its structure and functioning according to a defined point of view and with some detail level in order to improve the performance of this enterprise. This means that it is possible to have as much representation of existing views.

The study of different definitions presented previously has allowed us to define a process as a sequence of interrelated or interacting activities fulfilling the customer requirements which can be internal or external to the enterprise. A process can be triggered only by internal and / or external events, each of these events is a support of information; an activity receives and converts them into output elements (products / services) by the influence of control elements and using the required and available resources during a well defined period. Each process is in interaction with others related processes and can be structured into sub-processes depending on the desired level of detail, in other words: the definition of the same BP modelling, vary depending on the objective, we consider that this variation is due to the detail level in its description witch we can call the zoom level, this description is made on the basis of triggers events and objectives.

The BP modeling formalisms are generally focused on a detailed description used of the tasks that will be performed by the actors. Such an approach not only limits the reusability of models, but can also exceed the initiative which is part of the actor skills, and a possible dimension of interaction and cooperation. In our approach, we propose to develop the traditional metadata models, while redefining the concepts of actor and activity. The realization of a service involves the use of the skills and adequate knowledge. These skills are producing an added value. The continue adaptation to the request involves learning and the continued development of skills. However, from a practical way, learning can be managed and controlled by a set of uniform skills. So the business, as a set of activities involving the skills and homogeneous knowledge is the basic unit for the BP structuring and for the continued adaptation to change.

The proposed meta-model is based on two alternatives: the decoupling between the BP conceptual description and the organisational choices. Secondly it is driven by a separation between objective and goal [7], which does not consider that an order is a part of the goal structuring, because the two concepts have not the same semantic level: the goal is in management system level, and must be able to be matched to the strategic orientation. To enable the process to reach the goal, different ways can be designed, and the choice is solidified by the activities of the detailed process. But at the functionality level the goal is either an outcome that would be generated by an activity, or a service that can be accomplished by an operating actor.

Figure 6 describes a generic meta-model around the concepts of goal, activity and actor.

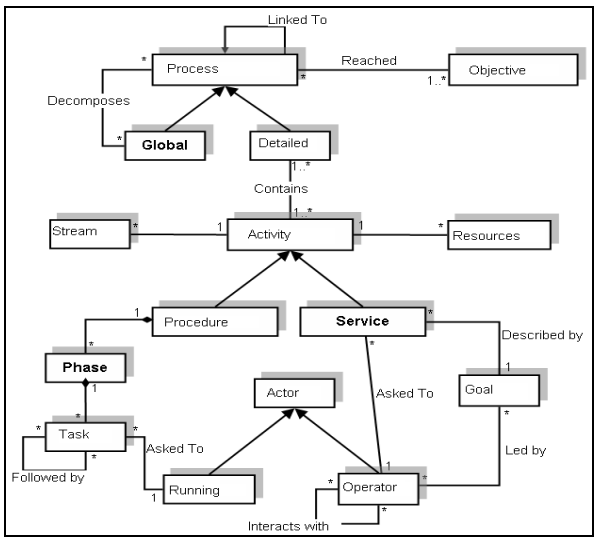

Figure 6. meta-model of a BP

All these concepts are defined as follows:

- Process: is a set of activities with the objectives of producing defined results.
- Actor: it is an element (individual, group, organizational or software entity) that is playing a role in the process definition.
- The Operating concept is defined as an actor with the ability to carry out activities without needing to indicate the operating mode, but only the goal.
- A performer is defined as an actor who has no autonomy and whose expectations are carrying out tasks in accordance with their description.
- The goal concept makes it possible to describe a purpose, more limited than that which is committed to the concept of objective.
- Activity: it is the functional unit of business process structuring. It defines the specific goal of an actor or action.

- Flow is a data exchange among activities that may be continuous or triggered.
- A procedure is an activity defined by the tasks that compose it and constitute its operating mode.
- Phase: it is a set of sequential tasks;
- Task: is an atomic set (indivisible) of actions chronologically ordered either in series or in parallel, triggered by one or more events related to a condition.
- The Service is defined by a goal; it will be assigned to a performer, with autonomy to accomplish it.

4.2. The MAS meta-model

In the BP modeling, the major objective of re-use was partially realized by the object approach. However, the initial objective of re-use was reformulated in terms of interoperability, adaptability, co-operation and social organization. Considering these concepts, the MAS technology is the best answer to these requirements.

The Meta models oriented agent: it represents agent concepts as well as their main relations and architectural properties.

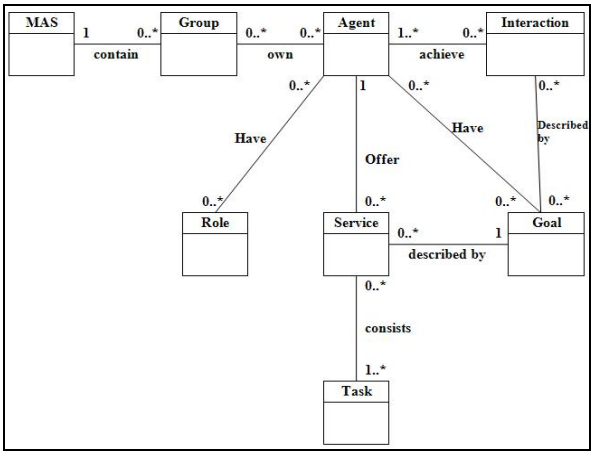

Figure 7. MAS Meta-model

The MAS meta-model can be obtained starting from the BP meta-model by applying mapping rules. Thus the BP mode is mapped into by a MAS model where the concept of the actor is represented by an agent. Activities are modeled within the limit of the goal and are controlled by an agent. If several agents contribute to the goal realization, the communications protocols represent a standard framework of this collaboration. In our approach a business process will be modeled by MAS. As an information system is a set of interacting BP, thus, it will be represented by a set of interacting MAS.

4.3. The approach Models

In this context, we use the UML models that represent the MAS; they are automatically mapped into object code (Figure 8):

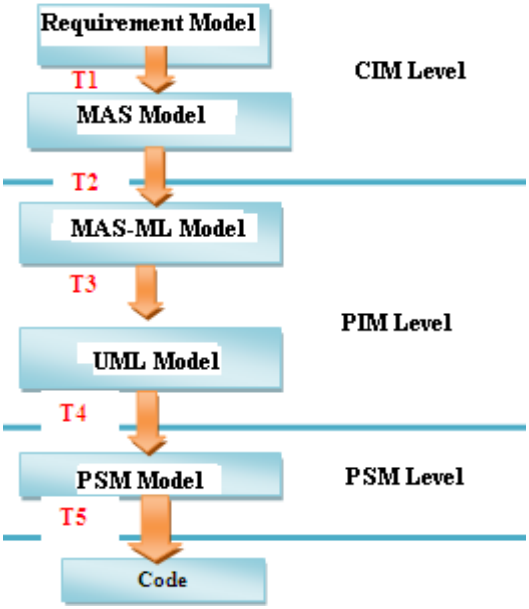

Figure 8. MAS development Process based on MDA

We will use several transformations (Ti) since the requirements model until the generated code.

5. Multi-agent Systems and MDA

5.1. An overview

Generally, the development process of the complex systems such as the MAS is extremely varied. It implies the development of various models based on various needs and constraints. The transition from an organization specification into an implementation via serial model transformations is often carried out in a badly organized way: nonsystematic and inadaptable with the technology evolution.

Currently the agent technology [16] becomes increasingly pervasive in various application domains, it becomes convenient and necessary to have a generic development process of the MAS, which clearly defines the various steps of the process as well as the rules and the tools necessary of transition from one step to the other and this in a systematic way. In this work, we propose a development process based on the MDA approach.

The MAS development process proposed is illustrated in the N° figure. The first step of the process defines the requirements capture while being based on meta-models: BP meta-model, MAS meta-model and the description of the MAS meta-model in MASML concepts. The CIM is defined by instantiations of these various meta-models.

Then the defined CIM models should be mapped into PIMs models independently from any specific platform such as AUML [20] and MAS-ML. In our approach, we propose the use of MAS-ML language. The principal difference between MASML and other MAS modeling languages lies in the fact that MAS-ML integrates the concepts of organization, role and environment like generic classes. By defining these abstractions as generic classes, it will be possible to define their, properties, relations, behavior and interactions.

Once PIMs are defined, it is necessary to define PSMs relating to a given platform. To implement the MAS, several architectures, platforms, and framework, can be used. In our approach, MAS-ML models are transformed in UML models by using the ASF framework [21]. The choice of ASF framework is especially based on the fact that it allows an implementation of all the MAS-ML language concepts.

Other MAS implementation platforms do not allow the implementation of all the MAS components. As example to implement a MAS described in MAS-ML language on the JADEX platform [19], several new mapping rules are necessary to convert MAS-ML models into AUML models that would be very difficult since JADEX does not use the same abstractions as MAS-ML [17]. The last step relates to the mapping of PSMs into generated code. The target programming language will depend on the selected specific platform. In our case, the targeted language is java language since it is the supported language by ASF framework.

5.2. MAS-ML Language

MAS-ML is a language of modeling dedicated to the MAS modeling. It is an extension of UML which integrates the concepts of role, organization and environment described in TAO ontology (TAMIS, AGENT, and OBJECT) [19]. Thus, by using MAS-ML, it is also possible to describe not only objects, but also the agents, environments, organizations and roles which are basic concepts of agent technology.

The choice of MAS-ML for PIMs description is due to three principal concepts which it integrates. It is an independent modeling language from any platform and allows a modeling in conformity with the MOF [10]. This property facilitates the enormously development process since the meta-data exchange standard of UML in XML (XMI) [6] can easily be used and allows the transforming from the MOF to XML. Consequently, it can be adopted for model description in conformity with the MOF.

The modeling of the organization concepts, environment and of role is not taken into account in the other MAS modeling languages. Without these concepts, it is very difficult for example to describe the fact that different agents can play the different roles in different organizations.

6. Enhancement of the business processes modeling

In the BP modeling (and IS engineering in a general way), the major goal of reuse has been partially realized by the object approach. However, the original purpose of reuse has been redrafted more ambitious in terms of interoperability, flexibility, cooperation and social organization. Given these concepts, the agent paradigm is the one that best answers.

6.1. Analogy IS/MAS

Different works tried to bring closer the MAS modeling and IS modeling, in different perspectives.

Some of them proposed models to represent MAS, like Kishore [2] who makes a large literature. In this perspective, the only actors are intelligent agents, and in general, the human actors are considered to be users of the system. Others used modeling languages of software engineering, notably UML, to represent such systems [3] [4].

Others tried to model communications among organizations, particularly by using the contract-net concept [5] [6]. Others proposed a business process vision as a conversation [7] or a succession of communication cycles between a customer and a purveyor [8], using the theory of the negotiation. Kishore [2] proposes a uniting frame between the MAS modeling and the modeling of system coordination, allowing to model business processes.
Wagner [9] proposes a meta-model based on differentiation among the agents, active elements, objects, passive elements, and oriented to the communications representation.

6.2. An oriented agent Meta-modelling of BP

We chose to represent the BP meta-model, introduced before, by the Agent paradigm and multiagent system (MAS). The MAS Meta models are obtained from Meta model domain by transformation rules.

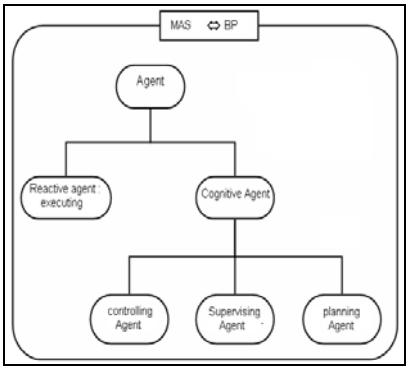

Figure 9. The BP Multi agent modelling

So BP is represented by MAS where the concept of actor is represented by an agent. Activities are modeled in term of purpose, and are putted under the supervision of an agent. If several agents are charging up of activity, the protocols of social convention kind represent a standard framework of this collaboration.

The BP agent modeling that we propose is based on a classification of the agents in five categories: Supervising agent, planning agent, controlling agent, executing agent and synthesizing agent. The figure 9 illustrates the MAS structure.

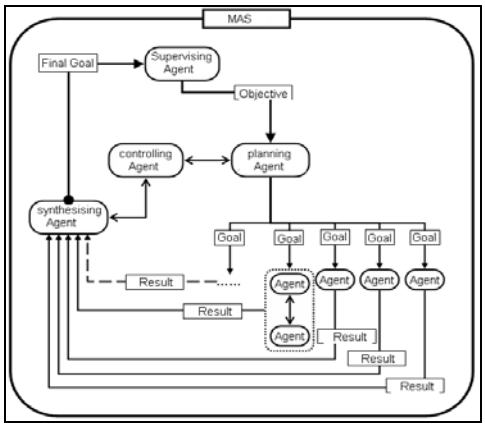

Figure 9. Agent Meta-modelling of BP

- **Supervising agent:** it is elected to achieve an objective that represents the finality of the BP, it is a cognitive agent having the capacity to define a formal representation of its objective, it transmitted this representation to a planning agent, which will take over, and at the end it controls the level of initial objective realization.

- **Planning agent:** that is also a cognitive agent that the role is to translate the description of the objective in goal, each of these goals is reached by an activity; firstly it established the activities sequence that it must reached, secondly dispatches these activities among the executing agents, this assigning is fully developed in consultation with a controlling agent.

- **Controlling agent:** its role is to make sure of the correct sequencing of activities, at any instant it knows the executing agents situation. This is the reason why it contributes to the planning.

- **Executing agent:** generally, encapsulates the totality of the tasks that compose an activity. Thus it is defined by the goals which it can reach.

- **Synthesizing Agent:** it is an agent that the role is to combine in a final result the outfit of the obtained results by the different executing agents, according to the pre-established order, and then this result is transmitted to the supervising agent to prove its compliance with initial objective.

6.3. Enterprises structuring and MAS rapprochement

Generally, the enterprises are structured round of a divisional structure (also called BU: Business unites), it structures the enterprise into unities, according to the business processes. These unities are also structured according to a functional designing that makes a BU composed of several functional unities FU, among which each one intervenes in BP. The IS designing of an FU participating in the process of an BU, can be more or less traced, reused on another UP.

We determine a mainly structure adequacy between the divisional structure of an enterprise and the multi-agent organizations. There is in both cases a structuring of the organization by units following objectives, but it is not sufficient to lead towards the IS specification based on cooperative BP. For this, we propose that the multi-agents organization should be structured according to Agent Hierarchy and MAS Hierarchy in contrast with a simple hierarchy. That's means with objectives communication rather than orders. Communications are made according to Agents Hierarchy (they are then filtered during passages among different levels), or between agents and MAS from the same layer.

The multiagent architecture constitutes the architecture to implement; it is obtained by transformation rules.

7. Visual Agent environment

Visual Agents is a Java based development environment allowing an oriented agent development process. It uses languages MAS-ML and ASF and based on three components: a graphic tool, to support the described models in MAS-ML, a mapping tool which transforms the generated artifacts by the graphic tool into the UML-XMI concepts and a generation code tool. The Figure 10 illustrates the Visual Agent architecture.

The graphic component carries out the MAS-ML diagrams and takes into account the first transformation generating a MAS-ML XMI file.

The mapping tool is the most important component; it transforms the MAS-ML XMI file into UML XMI file. The last component has in charge the java code generation starting from the XMI file.

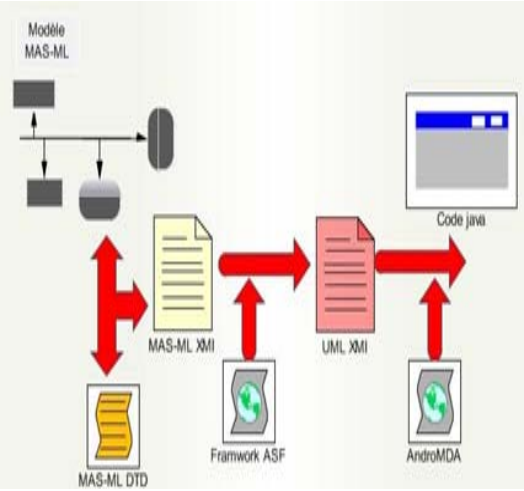

Figure 10. Visual agent architecture

9. Conclusion

We presented a multiagent approach based on the MDA and using a business process metamodeling for the information system development. Thus it uses several metamodels and models to transform abstract representations of an organization into an operational information system. This approach can be considered as an Interesting and promising reference framework which improves the information system development process. It is also aimed by Business Processes allowing to optimize and to ameliorate, the piloting of complex business processes.

The proposed MAS development process consists of four stages; the first step consists of defining the domain requirements, and states the MAS components, and the second step define the MAS-ML model of the MAS. The third phase consists in mapping MAS-ML models UML models. In the fourth stage, the UML models are automatically transformed into java code.

However, experiment with real case study is necessary to validate the suggested approach. For this, we are working on an e-business application that would be our next paper.

10. References

[1] Publication of SchlumbergerSema Consulting, pole « strategy, town planning and architecture of the systems of inquiry».

[2] Kishore R., Zhang H. & Ramesh R., 2004, Enterprise integration using the agent paradigm: foundations of multi agents-based integrative business information systems, Decison Support Systems.

[3] Paik I., Takami S. & Watanabe F., 2004, intelligent support loss in value agent supply in design based chain one services web semantic, Proceedings the of 4th international Conference one Hybrid intelligent Systems.

[4] Bauer B. & Odell J., 2004, UML 2.0 agent's |and|: how to build agent-based systems with the new UML standard, Engineering Applications of Artificial Intelligence.

[5] Weigand H.et van den Heuvel W.J., 2002, Cross-organizational workflow integration using contracts, Decision Support Systems, Volume 33, Issue 3, pp. 247-265.

[6] Hoffner Y., Lwdwig H., Gulcu C. & Grefen P., 2000, processes complicated cross organisational heart architecture year, Advanced descended E of-trades Web and-Based Systems inquiry. WECWIS.

[7] Aurimaki E., Lehtinen E.and Lyytinen K., 1988, to speech-act-approach modelling office |based|, ACM transaction one Systems inquiry office, 6, 2.

[8] Mentzas G., Halaris C. & Kavadias S., 2001, and Modelling business processes with workflow systems: of evaluation year alternative |approaches|, international management inquiry |of| journal, 21, 123-135.

[9] G. Wagner, 2003, The Agent-Object-Relationship Metamodel: towards a unified view of state and behaviour, Information Systems, 28, pp. 475-504.

[10] OMG (2001), MDA specification, MDA Guide V1.0.1 V. 1.0.1, http://www.omg.org/mda/ (Access date: April, 10, 2008).

[11] StarUML 5.0 Developer Guide, 2005, http://staruml.sourceforge.net/docs/developer-guide(en)/, (Access date: May 5, 2008).

[12] Java Agent Development framework (2000). http://sharon.cselt.it/projects/jade/, (Access date: June 6, 2006).

[13] Agent UML (2001), the FIPA Agent UML Web Site – http://www.auml.org, (Access date: June 10, 2007).

[14] Selma AZAIEZ thèse d'université de Savoie, Approche dirigée par les modèles pour le développement de systèmes multi-agents, 2008.

[15] Beatriz A et al., 2005. Visual Agent: A Software Development Environment for Multi-Agent Systems

[16] Jennings, N., Wooldridge, M.: Applications of Intelligent Agents. In: Jennings, J., Wooldridge, M. (Eds.), Agent Technology: Foundations, Applications, and Markets (1998): 3-28.

[17] Silva, V. and Lucena, C., 2004, "From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language". Journal of Autonomous Agents and Multi-Agent Systems 9(1-2), Kluwer Academic Publishers, p. 145-189.

[18] Silva, V., Garcia, A., Brandao, A., Chavez, C., Lucena, C., Alencar, P.: Taming Agents and Objects in Software Engineering. In: Garcia, A., Lucena, C., Zamboneli, F., Omicini, A, and Castro, J., (Eds.), Software Engineering for Large-Scale Multi-Agent System, LNCS, Springer-Verlag, 2003.

[19] L.Braubach, A. Pokahr, JADEX Tutorial (2007),

http://vsis-www.informatik.uni-hamburg.de/ projects/jadex/, (Access date: June 15, 2008).

[20] Silva, V. T.; Choren, R.; Lucena, C. J. P. a UML Based Approach for Modeling and Implementing Multi-Agent Systems. The 3th International Joint Conference on Autonomous Agents and Multiagent Systems, V. 2, (AAMAS'04), 2004.

[21] De Maria, B.A., Silva, V.T., Lucena, C. J. P. Developing Multi-Agent Systems Based on MDA. Proceeding of the 17th Conference on Advanced Information Systems Engineering (CAISE'05), Porto, Portugal, June 13-17, 2.

