International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-6, August 2013

oBDI2Jadex: An agent model based on O-MaSE methodology to design a BDI agents for Jadex

Jamal BERRICH, Toumi BOUCHENTOUF, Abdelhamid BENAZZI

Abstract -AOP agent oriented programming [3] is a new paradigm that is in a world of technological intelligence, the aim of this new aspect of development is to design robust and autonomous systems whose processing is distributed software entities called agent . The BDI agent is a particular type of agent based on the interaction with the environment to achieve specific tasks.
Currently, there are several containers that manage the life cycle of agents and especially the BDI agents [1] which is part Jadex.
Our goal is to normalize the creation of BDI agents adopting a design methodology called O-MaSE [2] by creating a new model to generate subsequently the agent system to be executed in the container Jadex

Index Terms—AOP, MAS, BDI, Jadex, meta-model, O-MaSE, Agents, aT3, EMF, XML, Ecore.

I. INTRODUCTION

The Organizations have increasingly used the computerized components of their information system makes it possible to offer services organized, reliable and quick access. We can see then that when it comes to saving time and money, the agencies do not skimp on the means regardless of the sector.

In this perspective, the idea is to create a model including BDI agents based on a design methodology and development to produce a body of active components to perform a number of tasks.

The realization oBDI2Jadex two main phases, the first realization of the model by respecting the meta-model O-MaSE [2], then the implementation of a code generator to create the employees' organizations to be deployed in the container Jadex [7]. Our achievement is a new contribution to all that is agent-oriented programming (AOP) [3],[6], it is actually the first application that provides a graphical interface for creating BDIS agents following the O-MaSE methodology and proposing generation all required for the operation of the agents in a container file as Jadex agents. OBDI2Jadex project development requires to know that the O-MaSE methodology and the basics of Java programming to fill its established multi-agent system.

Manuscript published on 30 August 2013.
***** Correspondence Author (s)
Jamal BERRICH*, Department of Computer Science Mohammed First University/ENSAO, Oujda, Morocco.
Toumi BOUCHENTOUF, Department of Computer Science Mohammed First University/ENSAO, Oujda, Morocco.
Abdelhamid BENAZZI, Department of Computer Science Mohammed First University/ENSAO, Oujda, Morocco.

II. BDI ARCHITECTURE

The BDI architecture is designed on the basis of the model "Belief-Desire-Intention", the rationality of an intelligent agent, developed for programming intelligent agents. This model is based on the implementation of the beliefs of a rational agent, cognitive, desires and intentions, he uses these concepts to solve a particular problem. In essence, it provides a mechanism to separate the activity of selecting a plan (sequence of actions) of its execution. Therefore, the BDI agents [1], [5] are able to balance the time spent deliberating on plans (choosing what to do) and the implementation of these plans (to do so).

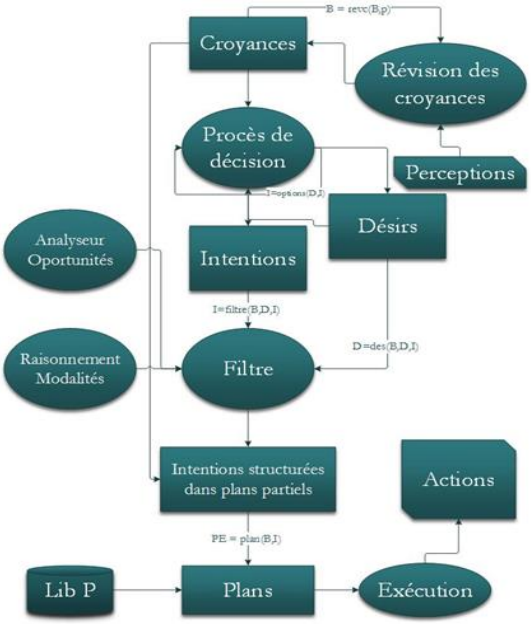

Fig 1: Architecture of a BDI agent

III. MATH O-MASE METHODOLOGY

A. Presentation

This method considers the system as an organization of agents, each agent has a specific role according to his ability to reach and achieve its objectives. So the purpose of this method is to build an organizational society agents based on the meta-model of the organization.

O-MaSE [2] is defined by the OPEN Process Framework, this framework allows us to produce a modeling methodology, it uses three concepts: meta-model, Method Fragments and Guidelines.

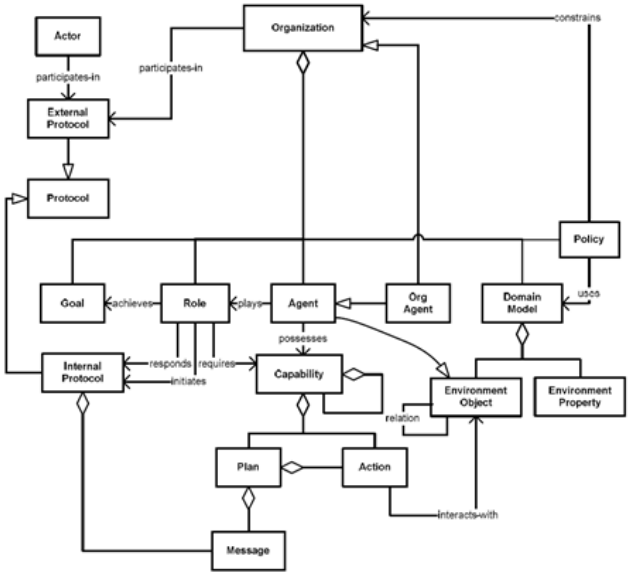

Fig 2: The meta-model O-MaSE

Goal : is an objective the system must obtain or satisfy. It is different from the function of the system: the goal is concerned that the result, however, the function looks at how to achieve the goal.

Role : is an entity capable of getting goals in the organization. A role must address at least one purpose of the system. To do this, he must have the responsibilities, rights and relationships. So there is a relationship between roles and goals.

Agent: agents play a / the role (s) and assume the responsibilities of roles with the use of its rights and relations. It is is an entity that perceives and can perform actions on the environment.

Capabilities: is the ability of an agent to deal with missions. Based on capacity, specifically the relationship between agents and roles are determined: the agents' own capacities and roles require some ability to play. So when an agent has the capacity required by a role, he can play that role.

Plans: The plans have a relationship with access and control of specific resources, the ability to communicate with other agents, the ability to migrate to a new platform, or the ability to use computer algorithms to the agent to perform specific functional calculations.

Actions: Actions are associated with hardware or software agents, they allow agents to perceive the real environment.

Domain Model: The real world is captured with the domain model. Basically, the domain model describes the objects in the environment (agents include) and the relationships between these objects (properties of the environment).

Policies: A policy describes how an organization can behave in a particular situation. Moreover, the domain model is also used to define specific organizational policies.

Organization Agents: are organizations that operate as agents of a top-level organization. LOs can play roles, have the capacity or coordinate with other agents. They added the idea of the organizational hierarchy of the meta-model of O-MaSE.

Protocols: This concept is included to capture the interactions between the organization with external actors, agents and roles.

B. *O-MaSE models*

The organization as well as their interactions with other organization and with external stakeholders.

Model Domain: Capture the domain entities, attributes and relationships between them.

Model Agent Classes: The objective of this model is to identify the type of agents that participate in the organization.

Model Protocol: Defines the interactions between agents and / or the actions performed by the agents on the environment.

Model Plan: Sets plans officer. a plan can be considered an algorithm to achieve a specific goal. In addition, plans are made of several actions or sub plans and may require interaction with other agents.

Model Policies: it defines a set of formally specified rules that describe how an organization can behave in particular situations. They identify the rules and constraints of the system. For example, policies can strain the behavior of playing a role in relation to other agents of the organization or restrict an agent to play some role combinations agent.

Model Actions: Capturing interactions exerted by environmental agents.

Model Services: Includes activities of the agents, in the exercise of roles, through services to other agents / roles.

C. *aT3 tool*

AgentTool III [2],[8] is an Eclipse plugin, enabling the modeling of SMA, it provides diagrams to represent models of the O-MaSE, it also allows you to:

1) Check the state diagram for each task or each conversation that is consistent is that each state has requirements inputs and outputs conditions;
2) Check the validity and matching different patterns of organization;
3) Automatically generate code from the model developed agents. This is the preliminary code on the JADE platform, the programmer must complete the following.
4) The development environment aT3 actually includes four components that are integrated into a single tool. These components are:
5) The graphical editor: It is the essential component of aT3 it supports graphic editing of each O-Mase models.
6) AT3 Process Editor: It enables the creation, management and verification of custom O-MaSE process.
7) The Verification Framework: The Framework audit AT3 provides a means to maintain consistency between their O-MaSE models using a set of predefined rules designers.
8) Code Generation: This component takes all design models created during the development in order to convert code that correctly implements the model. Obviously AT3 contains a Framework of automatic code generation. The only platform supported is JADE.

IV. oBDI2JADEX REALIZATION

The oBDI2Jadex architecture boils down to creating an eclipse plugin provides a graphical editor with a palette containing various elements modeling following SMA O-MaSE and a drawing area to represent modules. And a code generator for the Jadex agent container.

A. *Process of Use*

The process of using the oBDI2Jadex is:

task 1	After installing the plugin, the developer must create a new project in the dev environment, defining the project name and model.
task 2	The modeler, using a graphical editor, slide the different elements of the O-MaSE methodology from a palette into a drawing space to model the SMA
task 3	A BDI agent [4] is an agent who has a plan and a description file (ADF), for this it is necessary to generate these two files for each agent model in the second task, the files should be generated with the appropriate platform Jadex.

Tab 1: The oBDI2Jadex process

We can explain the process of the system by the following scheme:

Fig3: Processes Occupation oBDI2Jadex

B. *Achievement plugin for oBDI2Jadex*

The main objective of the plug-in to model and develop models of agents that will be integrated later into business applications according to O-Mase specification.

It is designed to be a complete development environment where the developer focuses on agent-oriented modeling.

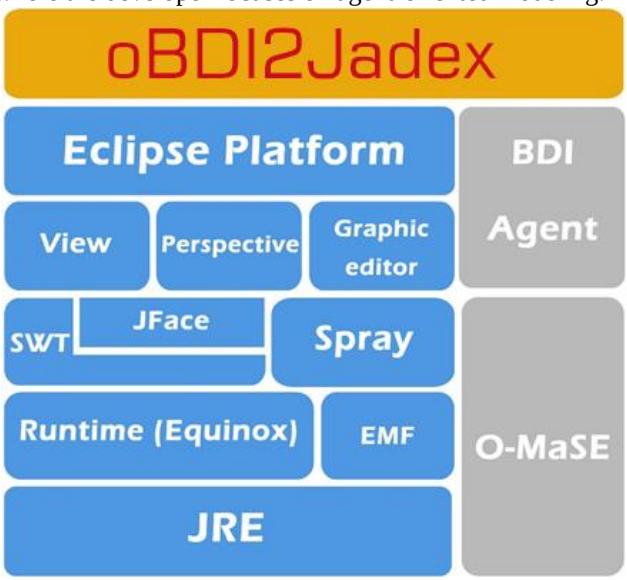

Fig 4: Structure of the eclipse plugin for oBDI2Jadex

Modeling plugin can be divided into two major parts, namely: modeling meta-model validation following graphs O-MaSE and instantiating meta-model using EMF technology.

The meta-model takes as its starting point the element of O-MaSE organization which plays the same role in Jadex package [7]. This organization (or package) is composed of several agents. Each agent plays a role. And a role can complete the goals and initiate protocols. The architecture is based on agents in Jadex goals, and each role can complete one or more goals. In the case of multi-purpose choice will be not conditioned deliberations and conditions.

As Jadex and O-MaSE, an agent has an ability that brings together a set of plans executed whenever their triggers are captured in the environment. He finally comes the Parameter element, specific Jadex, which allows you to add properties and other information relating to a plan, a goal or protocole.D the other hand, we noticed that significant elements in Jadex are absent in the O-MaSE metamodel as beliefs, facts and parameters. These elements are specific to the agent and differ from one agent to another. The DomainModel element O-MaSE is used to classify.

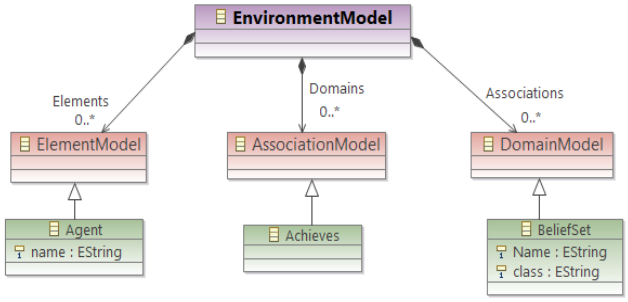

Fig 5: Relation between the components of the Ecore meta-model

1) *DomainModel*: This class is the parent class of all graph elements. All elements of the graph must inherit from this class.
2) *AssociationModel*: This class is the superclass of all the elements of such combination.
3) *ElementModel*: This class is the parent class of all O-MaSE or Jadex elements.

Item	Attribute	Description	possible values
Organisation	Name	The name of the organisation	string
Agent	Name	The name of the agent	string
Goal	Name	The name of the goal	string
	Type	The type of goal	achievegoal, performgoal, querygoal, maintaingoal
Condition	Language	the language of the condition	JCL
	Type	the type of the condition	Creationconditi on, dropcondition, recurcondition
	Content	The content of the condition	Code in JCL

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

capability	Name	The name of the condition	string

Tab 2: Definition of some properties of the meta-model

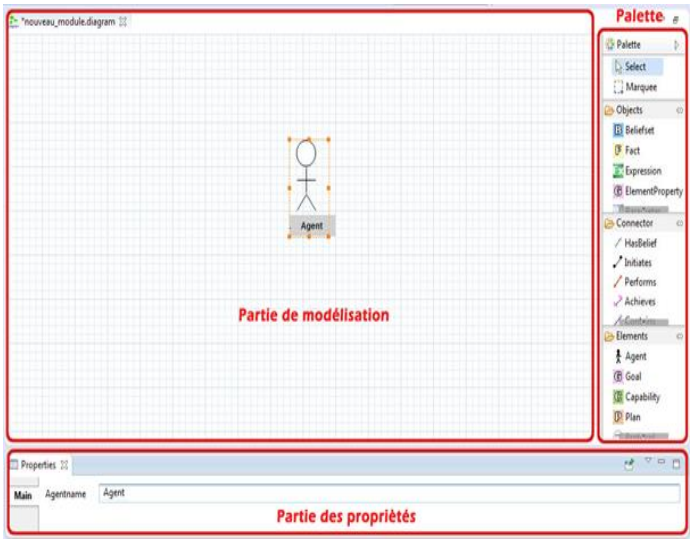

Fig 6: Perspective of the oBDI2Jadex project

REFERENCES

1. Anand S. Rao and Michael P. George,BDI Agents : From Theory to Practice, April 1995.
2. Scott A. DeLoach and Juan Carlos García-Ojeda, O-MaSE: a customisable approach to designing and building complex, adaptive multi-agent systems, Int. J. Agent-Oriented Software Engineering, Vol. 4, No. 3, 2010.
3. Ingrid Nunes, Carlos J.P. de Lucena, Uira Kulesza, and Camila Nunes, On the Development of Multi-agent Systems Product Lines: A Domain Engineering Process, AOSE 2009.
4. Ingrid Nunes, Simone Barbosa, Michael Luck, and Carlos Lucena, Dynamically Adapting BDI Agent Architectures based, AOSE 2011.
5. Busetta, P., Howden, N., R¨onnquist, R., Hodgson, A.: Structuring BDI agents in functional clusters. In: ATAL '99. pp. 277–289 (2000).
6. Kiczales, G., Lamping, J., Menhdhekar, A., Maeda, C., Lopes, C., Loingtier, J.M., Irwin, J.: Aspect-Oriented Programming. In: ECOOP 1997. vol. 1241, pp. 220–242. Springer-Verlag.
7. Pokahr, A., Braubach, L.: Jadex user guide. Tech. Rep. 0.96, University of Hamburg, Hamburg, Alemanha (2007).
8. Scott A. DeLoach and Mark Wood, Developing Multiagent Systems with agentTool, Intelligent Agents VII. Agent Theories, Architectures, and Languages - 7th.International Workshop, ATAL-2000, Boston, MA, USA, July 7-9, 2000, Proceedings,Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2001.