Implementation of Multi-Agents System to Control Adaptability in Workflow Environment

Hamdane Mohamed El-Kamel[1], Lezzar Fouzi[2], Boufenar Chaouki[3], Mili Seif Eddine[4]

[1] Computer Science Department, University of Khenchela, 40000, Khenchela, Algeria
[2,4] LIRE Laboratory, University of Constantine, Route Ain El-Bay Constantine, 25000 Algeria
[3] Computer Science Department, Jijel University, BP 98, Ouled Aissa Jijel, Algeria
[1]Hamdane.med@gmail.com, [2,4] fouzaw,seif03051982{@yahoo.fr}
[3] boufenar@hotmail.com

Abstract. The workflow process is often executed in a dynamic environment. This dynamic is classified in several aspects. For this reason, several researches try to give workflow architectures more adaptability in order to provide his business process more stability. This paper presented an architecture based agent to take in charge adaptability in workflow environment. The system proposed has the possibility to change their behavior according to the situation. To show the validity and feasibility of this architecture, it has been implement using Jade platform.

Keywords: Agent, MAS (Multi-Agents System), workflow, adaptability, Jade.

1 Introduction

The appearance of a workflow system in 1995 has lead most companies to integrate this technology into their infrastructure for the reason of its benefits. The WfMC defines a workflow as distributed multitask activity, routinized or systematized in some way, that involves the coordinated execution of human and system tasks, usually in heterogeneous environments [1]. We can see simply that the workflow is the automation of the business process[1]. The environment of the workflow is characterized by set of aspects as: fort communication, high degrees of coordination, distributed tasks, dynamic process, influence of the NTIC (New Technology of the Information and the Communication) …Etc. In environment as this it is necessary to answer the question “how to take in charge the stability of workflow system?

The management of adaptability in the workflow environment by using the agent[2] approach is a candidate solution [2]. So, we can see a workflow environment as a

---

[1] The process is a set a tacks
[2]Agent: encapsulated computer system, situated in some environment, and capable of flexible autonomous action in that environment in order to meet its design objectives [8].

distributed system that is focused on the coordination. This coordination is also based on a high density of communication. For example: if a customer has lost its credit card, it is strongly advised that retrieves another in sort time if the bank wants to keep this client. The treatment of this event (adaptable case) should be automatic after a louse's declaration.

The paper is organised as follow. Section 2 reviews some related works on the idea to taking in charge adaptability in workflow environment. Sections 3 provide short presentation of our architecture. An implementation prototype is presented in Section 4 and a case study is provided in section 5. Section 6 contains the final remarks and a summary of the paper.

2 Related Works

Several researches works as [3] [4] [5] try to contribute for the problem of the integration of adaptability in a workflow system using the agent[1] approach. After student of these architectures, we have concluded that each architecture is implemented in way to treat one aspect of the adaptability. For example architecture [3] is designed in a way that gives advantage to the care of adaptation level scheme workflow. But on other hand, the two other architectures [4] [5] are oriented towards the management of adaptability in the instances level.

Among the disadvantages of these architectures is which cover only one aspect of the adaptability. But, we concluded that this concept is located has several levels in an environment workflow [6]. We note that there is no clear classification of adaptability's cases which cause some problems during treatment.

The main drawback of these architectures is that they are trying to achieve support for adaptability by the idea of specialization agents, i.e. in all the agents of the architecture goes on sacrificing an agent(s) to achieve adaptability in the workflow environment [6].

Therefore, our effort will be focused firstly to propose a MAS that model all aspects of workflow system (as functional, behavioural, organizational, informational,). Secondly, ensures that this architecture is able to modify these behaviours according to the situations in order to preserve maximum the stability of workflow process.

There are few research efforts in the direction of integrating the concept of adaptability in workflow environment. In this work, we present the prototype of architecture which uses the paradigm MAS to cover adaptability in all levels of workflow system.

3 The global architecture

In this section we present the great ideas of our architecture. Our step is to propose firstly a generic strategy for taking in charge adaptability in a workflow system. This strategy is based in MAS paradigm as we have proved by another work in [6]. In summary, the strategy proposed defended two levels of adaptability; i.e. global level and local level. It is implemented in each agent in the form of rules, it east of **Fig.1** are stores in a base named “***Base Rules for local adaptability***” and “Base ***Rules for global adaptability***”. Its functionality is similar to a football match! The coach developing a tactical to play and each payer does utmost to follow this tactic. If a player came out by red card for example, the group should therefore follow their tactics in adaptable mode.

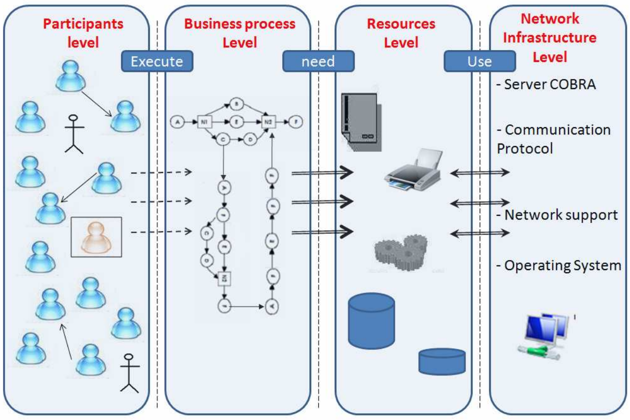
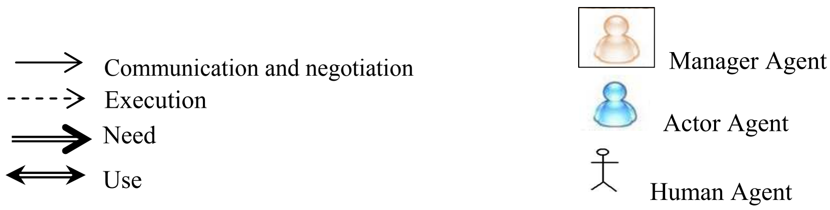


Fig. 1. Multi-Agent Architecture of workflow system



Hence, in order to provide the proposed architecture by the necessary flexibility, we propose that each actor Agent provides three missions. Firstly, each agent of this type achieving some tasks in the workflow process when we call principal role. Secondly, it can guarantee a few secondary roles in workflow process. Thirdly, he is responsible to managing the adaptability in his local environment (local adaptability). In architecture like this, modification of the scheme in local level of an agent actor doesn’t require a broad restructuring of the overall scheme of the process workflow.

An activity represents a set of roles where an agent can ensure in workflow process (part of process). I remember that exist tree types of activities: automatic activity, semi-automatic Activity and manual activity. The agent actor can only execute the automatic and semi-automatic.

Concerning the role of the manager agent is to make sure adaptability in global level [6]. We propose the sequence diagram of AUML method [7] to illustrate the functioning of the architecture **Fig 2.**

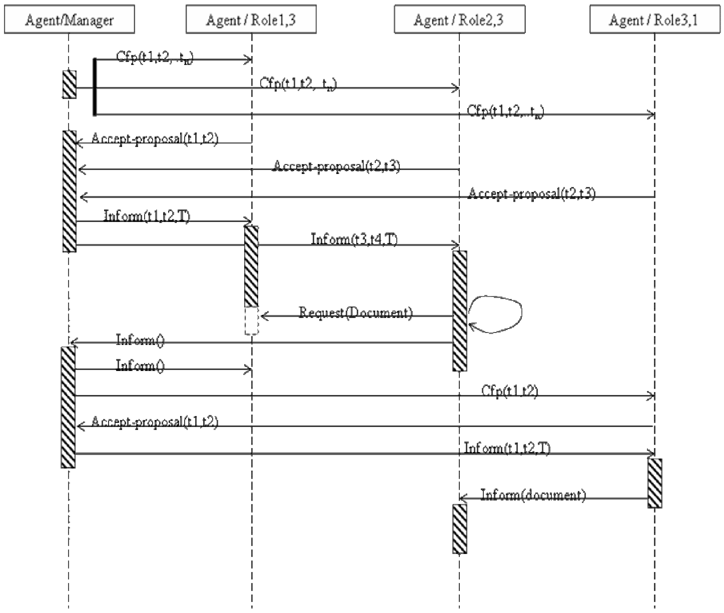


Fig. 2. Example of the Sequence diagram between SMA

4 Realization of the purposed architecture

4.1 Communication in the Agent-based Architecture

The agents need to communicate among these, as well as with the services of their platform or environment. Several mechanisms of communication are possible: message exchanging and sending, methods invocation and the use of the 'blackboard' mechanism.

Consequently, standardized inter-agent communication languages should be supplied. KQML (Knowledge Query Management language) [8] was proposed to support inter-agent communication. This language defines a set of types of messages (called "performatives") and rules, which define the suggested behavior for the agents that receive these messages. This language was developed in ad-hoc way for the requirements of the developers of agent's software packages.

The ACL (Agent Communication Language) [9], successor of KQML, supplied a richer semantics. This language was proposed by FIPA [9], which aims to standardize communication among agents. ACL is also based on language theory and is close to KQML at the level of the acts of the language, but not at the semantics level, which underwent a big improvement in ACL.

In our approach, we use the ACL to formulate messages and the XPDL (Xml Process Description Language) to describe the continents of messages. The use of ACL-XPDL in inter-agents communications permits to achieve a first of interoperability by surpassing the problem of heterogeneous exchanges among the different actors in the workflow environment.

4.2 The exchanged messages

FIPA [9] supplies a standard syntax for messages. These messages are based on the theory of the act do speech, which is the result of the linguistic analysis of human communication [9]. The basis of this theory is to produce an action from the language.

In the FIPA-ACL, no specific language for the description of the contents of messages is imposed. Several languages can be used for the description of the continents of the exchanged message such as KIF (Knowledge Interchange Format), Semantic language (SL), prologue and XML (eXtensible Mark-up Language)

XPDL (based in XML language) will be used firstly for the description of workflow process because is one of the recommendation of WFMC [10]. Secondly this language is used for the specification and interpretation of the contents of messages exchanged among agents from the workflow environment. So, the messages exchanged in our architecture are described in FIPA-ACL/XPDL.

The use for XPDL for the contents of communications among agents gives the possibility to display of messages in a Web browser and facilitates the integration with others web-based applications.

The following example illustrates interaction between the Manager Agent and an Actor Agent1, Agent Actor2 during a phase of taking in charge of adaptability.

The Manager Agent announces the sub-goal “Realization of task t1” as followings:

```
(cfp
:sender Agent_Manager
:receiver Agent_Actor1,Agent_Actor2
:language XPDL
:protocol fipa-contract-net
:content ( <Activity Id='tack3' Name='Traitement de la
Declaration'>
          <Implementation>
          <operation1>
          <Condition type='CONDITION'>Cotisation==false</Condition>
          <Send Id='Lettre-refus'>Envoyer lettre de refus au
client</Send>
          </operation1>
          <Selection>Sélection un expert </Selection>
          <Send Id='File-Declaration'>Envoyer lettre de
mission<Send>
          </Implementation>
          <TimeEstimation>
          <Duration>48 heure</Duration>
          </TimeEstimation>
          </Activity>
:reply-with propose)
```

Based on the Contract-Net Protocol [5] the Manager agent, which is the initiator of the workflow process, asks various potential agents to tender their offers, i.e., to give the necessary information concerning their activities (role, resource, time…) in order that the manager Agent checks them.

Fig. 3 illustrates the process of communication between the Manager agent and an Agent actor. The communication process is support by "Communication component" [6] module in the manager agent. The FIPA-ACL message are formulated and sent by the communication module of the Manager agent and actor's agents.

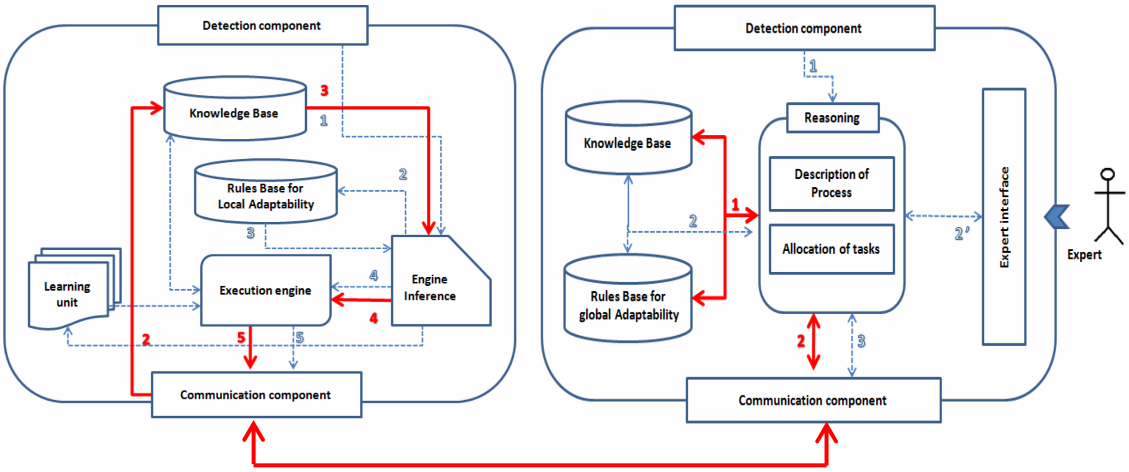


Fig. 3. Flux of communication between Manager Agent and Actor Agent 1

Ordinary situation of process communication

Process of communication in adaptability situation

5 Case study and simulation

We have simulated our approach in the domain of automobile insurance (see **Fig. 4**). In this case study; we have to approach the problem of the coordination and so the communication during the execution of the workflow process in order to take in charge adaptability if a change appear.

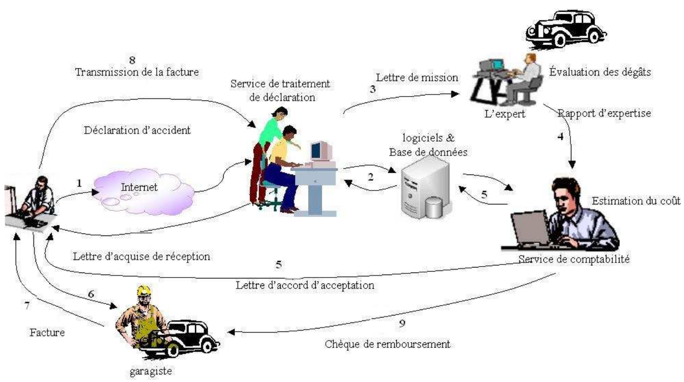


Fig. 4 process workflow of automobile insurance agency

5.1 General description of the workflow process

From the implementation viewpoint; we would adopt the use for an agent-based platform to implement the different agents and concepts of our architecture. Several platforms are supplied as software package; such as JADE [11]; ZEUS [12]; for the cognitive agents or SWARMW [13] for the reactive agents.

For modelling the workflow process of automobile insurance agency we have used the language XPDL because the WfMC classify it as a standard language for description the business process [10]. We have user the toolset of "Together workflow server community" [14] to model this process workflow:

```
  .........
  <Activity Id='tache3' Name='Traitement de la
Declaration'>
    <Implementation>
      <operation1>
        <Condition
type='CONDITION'>Cotisation==false</Condition>
        <Send Id='Lettre-refus'>Envoyer lettre de refus
au client</Send>
      </operation1>
        <Selection>Sélection un expert</Selection>
        <Send  Id='File-Declaration'>Envoyer  lettre  de
mission<Send>
    </Implementation>
    <TimeEstimation>
      <Duration>48 heure</Duration>
    </TimeEstimation>
  </Activity>
.........
```

5.2 Using Jade to implement MAS in the workflow environment

JADE (Java Agent DEvelopment framework) is the multi-agent platform; developed in Java by CSELT (Research Graouppo Telecom, Italy). It provides a FIPA (Foundation for Intelligent Physical Agents) compliant environment and execution of MAS [11].

JADE includes two basic constituents: an agent platform and software package for the development of agents in Java. It supplies several facilities, among them, we can mention the following [11]:

- A distributed agent platform: the agent platform can be spilt among several hosts (provided they can de connected via RMI). Only one Java [15] application, and therefore only one Java Virtual Machine is executes on each host.

  Agents are implemented as Java threads and live within Agent containers that provide the run-time support to the agent execution.

- An efficient transport of ACL message inside the same agent platform. In fact, messages are transferred encoded as Java objects; rather than strings. When crossing platform boundaries, the message is automatically converted to/from the FIPA compliant syntax, encoding; and transport protocol. This conversion is transparent to the agent implementers that only need to deal with Java objects.

- A library of FIPA that is ready to be used.

In Jade, an agent is an instance of the Java class defined by the programmer. This class itself is an extension of the basic Agent class (included jade.core). It implies the inheritance of the set of basic methods to implement the personalized behaviour of the agent. The agent is implemented using multitasking; where the tasks (behaviours) are executed concurrently. Every functionality supplied by an agent must be implemented in one or several behaviours.

The fallowing Java code represents the implementation of the Manager class and the Actor agent class. These classes are extensions of the basic Agent class respectively of the manager agent class and Actor agent.

```
public class Agent_Manager extends Agent {
protected void setup () {
addBehaviour (new Simple Behaviour (this)){
// Processing
}
}
public class Agent_Actor extends Agent {
class reception extends simpleBehaviour {
// Processing
}
Public reception (Agent a){super (a);}
protected void setup() {
reception mybehaviour = new reception(this);
addBehqviour (mybehaviour);
}
}
```

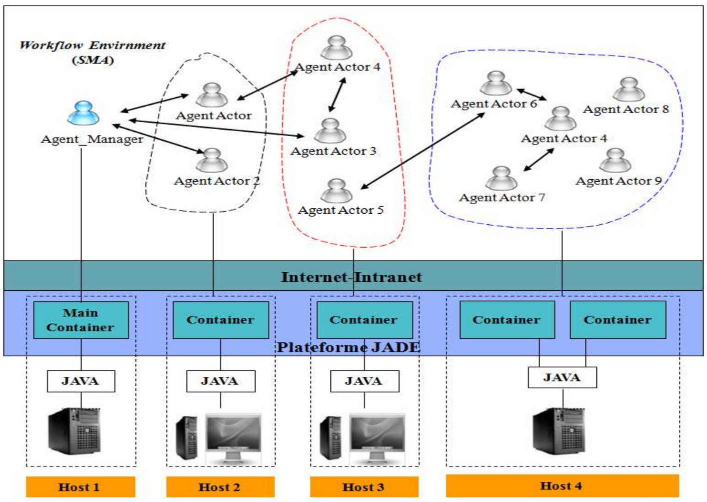


Fig. 5. Implementation of MAS architecture in JADE

5.3 Experiment results and consideration

In order to illustrate the main of our idea which has been introduced in sections 3 and 4 and to see the reflection which must be made by MAS to respect the strategy of adaptability implemented two classes of adaptability cases are simulated using the MAS.

It is supposed that Agent manager carries to starting the process workflow through CFP (call for propose) for the realization of the tasks (7 tasks) by using the protocol Contract-Net.

- The first scenario is the following: it is supposed that an actor agent is blocking during the execution of its activities. At the time when the process is running, the manager agent realized that the actor agent 3 is blocking. Hence, its return into *Base rules of global adaptability (BRGA)* to see what rule is adequate in the case (blocking of an agent actor). According to this base, it must applies the action defines by rule N°1 which indicates that it owes reassigned this tasks in other agent actor; i.e. provided that the agent actor support this tasks such as secondary tasks. In jade, l' Agent manager consulted the *DF* of the platform to recover a set of agents actors, he find the agent actor 5.

Thereafter, the manager agent sends to “agent actor 5” a CFP for the realization of tasks 5 and 8. The agent actor 5 answers favorably the offer of the Manager agent and it thus takes again the role of the agent blocked “Agent3” while the workflow process continues to function.

The figure **Fig. 7** presents the interaction between the manager agent and a whole of Actor agents to adjust the operation of the system caused by the blocking of the agent Actor 3”.

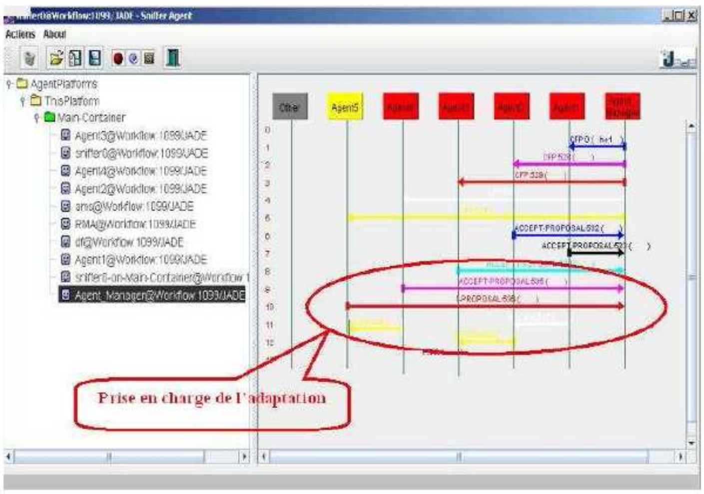


Fig. 6. The result of the interaction between SMA in scenario 1

- Scenario 2: the second example supposed that the change goes affects the reorganization of the workflow process. It is supposed that the company has to decide to modify its process by requiring a second expertise for any accident (add of new tasks). Thus, the expert goes introduced the description of new task in XPDL through expert interface (see **Fig.7**).

Manager agent made resorts to his "base rules of global adaptability" to see what the action which must be connects if a new task east injects into the diagram of the process workflow. Manager Agent must carry out the rule N°6 which indicates that the Agent Manager (in collaboration of course with the expert agent) goes and redefine the scheduling of the tasks of the workflow process and thus the allocation of the task " second expertise" with an agent actor of the system.

Fig. 7. Add of new task through the Expert Interface in Agent Manager

The following figure represent the process of take in charge adaptability when one adds the task “second expertise” in the diagram of the process workflow (**Fig. 8**).

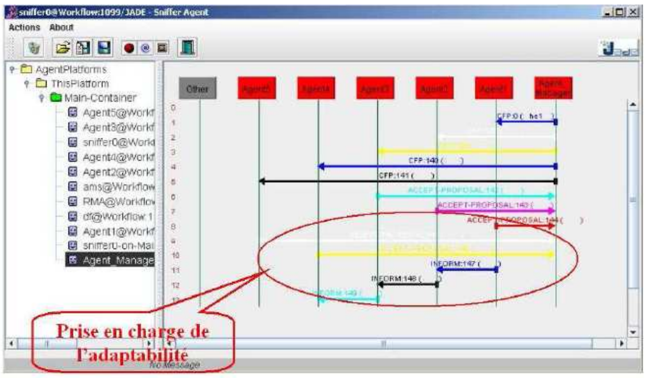


Fig. 8 The result of the interaction between SMA in scenario 2

6 Summary

In this paper we have tried to make the workflow more flexible by using the MAS paradigm. The strategy suggested and the architecture that it support has makes more flexibility in the system. Hence, the presented solution could guarantee a degree of stability of the process workflow in a rather dynamic environment.

We can conclude that the MAS approach can be used as methodology to make more stability of workflow process. Moreover, its concepts are adequate to taking in charge adaptability in a workflow environment. A prototype has been developed to show the feasibility and validity of this idea. Our future work will focus to cover all possible combination of adaptability cases and to achieve communication between MAS and others participants in external environment applications.

References

1. The Workflow Management Coalition, Workflow Management Coalition WorkflowStandard,” Process Definition Interface -- XML Process Definition Language”, Document Number WFMC-TC-1025, October 3, 2005, http://www.wfmc.org,
2. N.R. Jennings, P. Faratin, T.J. Norman, P. O’Brien, and B. Odgers, “ Autonomous Agents for Business Process Management ”, Journal of Applied Artificial Intelligence, 14(2), pp. 145–189, 2000.
3. N.C. Narendra “Flexible Support and Management of Adaptive Workfow Processes”, Information Systems Frontiers 6:3, 247-262, Kluwer Academic Publisher, 2004.
4. Robert Muller,Ulrike Greiner ,Erhard Rahm, “ AGENTWORK : a work.ow system supporting rule-based workfow adaptation”, Science Direct, Published by Elsevier B.V, 2004.
5. Lars Ehrler, Martin Fleurke, Maryam Purvis, Bastin Tony Roy Savarimuthu , “Agent-based Workflow Management Systems (WfMSs), JBees – a distributed and adaptive WfMS with monitoring and controlling capabilities”, Journal of Information Systems and E-Business Management manuscript 2005.
6. Hamdane Mohamed El-Kamel, “Toward a strategy based agent for the managing of the Adaptability in the Workflow Environment”, Journée Jeune Chercheur en Informatique JCI’08, université 8 Mai 54 Guelma, Algeria, 2008
7. Odell james, H Van Dyke Parunak , Bernhard Bauer, « Extending UML for Agents », Jamesb Odell Association, www.jamesodell.com, 2001.
8. Ferber J, "les systeme multiagents: Vers une intelligence collective" InterEdition, 1997, 239-245
9. FIPA; 1999, http://www.fipa.org/spec/FIPA98.html
10. The Workflow Management Coalition, Workflow Management Coalition Workflow Standard,” Process Definition Interface -- XML Process Definition Language”, Document Number WFMC-TC-1025, October 3, 2005, http://www.wfmc.org/standards/XPDL.htm
11. Fabio Bellifemine, Giovanni Caire, Tiziana Trucco ,« Jade Programmer’s Guide », university of Parma 200-2003, http://jade.cselt.it.
12. ZEUS 1999 , http://www.labs.bt.com/projects/agents/zeus
13. SWARMS 1996, http://www.swarm.org/index.html
14. Together workflow server community, http://www.together.at/index.html
15. http://www.sun.com/jini/whitepapers/architecture.html