

Network human-robot interface at service level

To Dong Nguyen* **, Sang-Rok Oh, Bum-Jae You *

*Intelligent Robotics Research Center - Korea Institute of Science and Technology, Seoul, Korea
(Tel : +82-2-958-5708; E-mail: {ntdong, ybj, sroh}@amadeus.kist.re.kr)

**Human Computer Interaction department - University of Science and Technology, Taejon, Korea

Abstract: Network human-robot interface is an important research topic. In home application, users access the robotic system directly via voice, gestures or through the network. Users explore a system by using the services provided by this system and to some extent users are able to participate in a service as partners. A service may be provided by a robot, a group of robots or robots and other network connected systems (distributed sensors, information systems, etc). All these services are done in the network environment, where uncertainty such as the unstable network connection, the availability of the partners in a service, exists. Moreover, these services are controlled by several users, accessing at different time by different methods. Our research aimed at solving this problem to provide a high available level, flexible coordination system.

In this paper, a multi-agent framework is proposed. This framework is validated by using our new concept of slave agents, a responsive multi-agent environment, a virtual directory facilitator (VDF), and a task allocation system using contract net protocol. Our system uses a mixed model between distributed and centralized model. It uses a centralized agent management system (AMS) to control the overall system. However, the partners and users may be distributed agents connected to the center through agent communication or centralized at the AMS container using the slave agents to represent the physical agents. The system is able to determine the task allocation for a group of robot working as a team to provide a service.

A number of experiments have been conducted successfully in our lab environment using Issac robot, a PDA for user agent and a wireless network system, operated under our multi agent framework control. The experiments show that this framework works well and provides some advantages to existing systems.

Keywords: human-robot interface, multi-agent, FIPA, Jade.

1. INTRODUCTION

Network human robot interface (HRI) is the interface among the partners in a robot system through the Internet to explore the service provided by this system. Partners in this system may include robots, distributed sensors system, internet information system and users that directly interact with the system via voice, gesture or through their digital representative such as user agents on a PDA. Each partner sees the system as a service provider and tries to explore this service for achieving its goal. In this paper, we are looking for a method that manages this service system.

The network HRI can be implemented by using a client-server model, a distributed model or using agent technology. The concept of network human interface varies from each implementation. Hideaki Takeda et al [1] introduced the concept of ubiquitous communication. Users may directly command the robot using voice, gestures. On the other hand, they can use the ubiquitous access. Users can make a gesture to a camera in the network and the command may be sent to the robot using the network system. They introduced an agent-based architecture to solve the above task. The main idea of this research is a ubiquitous access method that explores the agent as a means to communicate human and robots via the distributed sensors. In this method, users get no feedback from robot system as they communicate using distributed sensors. Users should be trained how to command a specific robot. Maxim Makatchev and S. K. Tso [2] proposed a framework using a proxy agent architecture. They try to solve the communication problems by using the proxy agent as a mediator between robots and users. Therefore the command may be store and forward even there is some

communication problem. This research deals with the offline situation. However they didn't solve the task allocation and service exploration problem.

Similar to the above researches, our research explores the agents for the HRI using distributed sensors and multi-agent communication method. Our research focuses on exploring the robot system at service level. It deals with the offline situation and provides the task allocation method for each services. In our network HRI approach, the world is considered as a set of workspaces. Each workspace is controlled by a multi-agent system. Firstly, this multi-agent system is used for fusing distributed sensors to monitor the environment. Secondly, it works as the center to dynamically coordinate with robots and users in the environment. When robots enter a workspace, they register their services, explore the information of this workspace and provide their ability to the system and users. Users use their PDA or notebook to access the system, discover and access the services provided by robots in the workspace. This system is able to organize, discover and explore the services of each partner.

Service discovery methods are very important in agent technology. In the agent-based standard like FIPA [3], service discovery in an agent platform (AP) is provided by a special agent named Directory Facilitator (DF). Agent can register, deregister and modify service information in DF. DF provides the service management mechanism to design an Internet HRI interface system. However, DF does not provide the ability to interact with the offline service. The presence of a robot in the system is not permanent. The robot may be offline at the time when the users want to assign the tasks, but it may be available later. Therefore, a mechanism to discover the services, assign the tasks to the robots, and provide the interface for service interaction when robotic agents are offline, is needed. Our framework supports a method to deal with the offline situation.

The responsive environment is the environment that can provide information about itself to the other partners working in it. Turning an environment into a responsive environment by installing sensors around the environment is one approach that is used in some human tracking systems, where a number of cameras are installed and connected together. K. Huang and M. M. Trivedi [4] proposed a system, in this the camera array is installed around the workspace for human movement tracking. In the implementation described later, the environment sensor information and local sensor information are combined by exploring multi-agent architecture. Developed from the idea of exploring a central control system to control a group of robots by changing the behavior of the whole group that proposed by Ali, K. and Arkin, R.C [5], our responsive environment can change the behavior of any robot placed in its environment to make this robot adaptive to the environment.

A complicated service needs an effective task allocation method to select which robots will coordinate together and to solve resource confliction. To do this task allocation we proposed the usage of coordinator agents for each service. This coordinator agent will coordinate the task using the information about the probability of being available at required time that can find in the VDF system and using a modified contract net protocol to select the coordination partners. Our task allocation service may provide several levels of service organization such as single user level, multi-user level.

2. MULTI-AGENT FRAMEWORK

2.1 Multi Agent Framework

In the Internet HRI, one problem is the offline situation, when a robot is temporally out of control due to re-charging of battery or suffering from network problems. Our framework is designed to solve this problem. Figure 1 describes the system with robotic, user and environment agents.

Robotic agent: The software is designed to separate two parts: the control part and the interface part. In the interface part, all behavior and services are arranged as agents. In this paper, only the Internet-based services are considered. The internet-based services can be categorized into some types and with each type the system processes the service in different ways.

The Internet human-robot interface services are services that require human interaction. With these types of services, robots create the slave agents for the services and migrate them to the main container. These services require no pre-defined algorithm and configuration. At the main container, user can manage, send command and work with the services.

The robot and environment interface services are services that require coordination with each other or with the distributed sensors. These services use the pre-defined coordination agents and algorithms to connect these agents together in a special protocol. The coordination protocol should take into account the possibility of communication problem.

Responsive Environment: The responsive environment is controlled by a main container and environment agents so that it can respond to the change in the environment. It gathers information from distributed sensors and works as the management part for the services of each robot in the environment. It also provides the interface of the services on

request. The coordinator agent should be located on the main container to coordinate robots and distributed sensors.

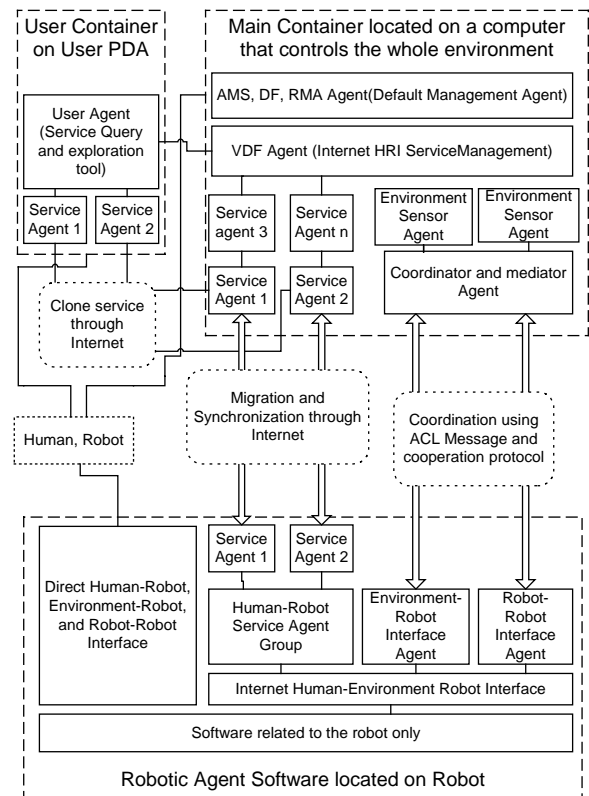


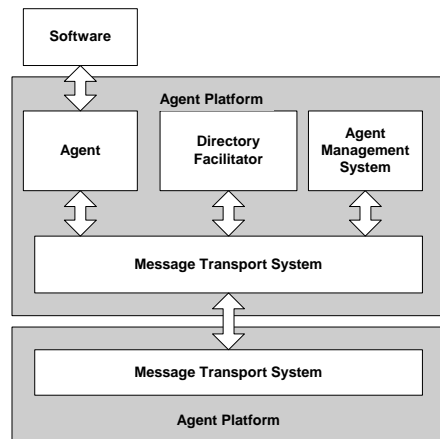
Fig 1. The overall framework of the proposed Internet-based HRI system

User Agents: Users access the system via a User Agent (UA) located on their PDA. On the PDA, there is no information about the robots. UA will query and clone the available services from the main container. With this UA, users can also interact with the services via ACL communication.

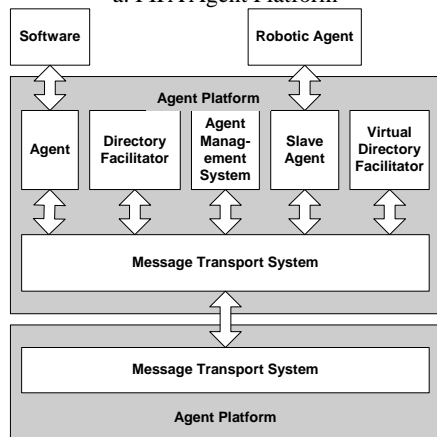
2.2 Virtual Directory Facilitator (VDF)

The FIPA standard is selected to implement the system. The central part in the FIPA standard is the definition of a management standard using a central control in the main container. In our framework, the implementation part and the interface part of a service are separated so that the interface part can be moved around the network by migrating and cloning. Each robot appears in the system as an agent. Each time this agent registers to the AP, it will create its service slave agents. Using the mobility feature of the AP, slave agents will migrate to locate on the main container. These agents work as slaves for the robotic agents. They are created at the time the robotic agents appear in the system and remains for a particular period of time that specifies by the robotic agents. The slave agents collect information when the robotic agents are offline, get the requirement or interact with users and other robotic agents. It can provide complicated graphic interface. Therefore, users with no experience about the robot can interact with it because after being migrated to the user device, it can be used as a normal application program. Then the slave agents synchronize with the robotic agent when it is online. Each robotic agent can visit many workspaces and drop slave

agents there. The existence of this slave agent is temporal. To control such a slave agent group, a special agent in the agent platform is created. This is called a VDF agent. VDF works similar to DF in FIPA standard but all of these services are provided by slave agents. VDF is responsible for synchronizing the slave agent and robotic agent. The UA located on the user's PDA works as a service query-and-display tool. UA can query all the services currently available on a system and then select the service that the user wants to use. After a service is selected, VDF will clone a copy of the service interface, move it to the container in PDA. After users interact with the interface, all data will be synchronized among interface agent.



a. FIPA Agent Platform



b. FIPA Agent Platform with VDF extension

Fig 2. Management architecture of FIPA standard and VDF extension

This system helps each robot to broadcast its service even after it deregisters from the system temporally. Users save the time for interaction with services because they can access the system at any time, work with the interface they find in the VDF and explore this service by invoking the interface. The availability of the system increases since the demand of long-term or permanent Internet connection is reduced and each robot only needs to be connected for a short time to migrate its service interface or to synchronize the data.

2.3 Robot-Environment interaction

A mobile robot normally uses only its local sensors. It is hard for robotic agents to work and cooperate in the unknown

environment because it does not know the change in the environment beyond its sensors' range and the existence of the other robotic agents. With a multi-agent system, our environment becomes a responsive environment. This multi-agent system helps to overcome these problems by using coordinator agent to coordinate distributed sensors with robot agents. By installing sensors around the workspace and connecting these sensors to the multi-agent platform, the information about the environment can be provided. Using this information, the responsive environment can discover the position of each robot and control the behaviors of a robot.

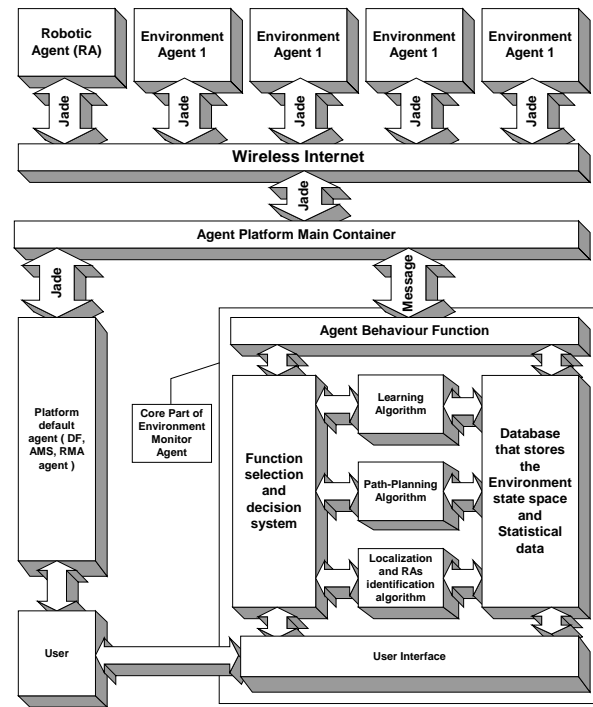


Fig 3. Users directly access the main container

2.4 Offline task allocation using Contract Net Protocol:

Using the VDF, users can access a list of services provided by the system. There are some services required the task allocation among some robots or network systems. To solve this task allocation we use the contract net protocol. Contract net protocol is a standard part in FIPA standard. However, our system works with the offline robot so the VDF should do more things than simply call for proposal of each part of a service.

A complicated service needs an effective task allocation method to select which robots will coordinate together and to solve resource confliction. To do this task allocation we proposed the usage of coordinator agents for each service. This coordinator agent will coordinate the task using the information about the probability of being available at required time that can find in the VDF system and using a modified contract net protocol to select the coordination partners. Our task allocation service may provide several levels of service organization such as single user level, multi-user level.

Some complicated services require the existence of several agents in the system. Moreover, some agents at the same time can contribute to the same part of a service at different level of performance and available level. If an agent is selected to

provide a service, it may not be available for other service. In the user interface level, users should be able to know:

- *All the service may be available:* We select all the usable combination of agents and use the offline contract net protocol to show the level of availability of each case. If users insist to do some services they can try to explore this service level. In that case when a service is selected, all the available services should be calculate again. If a service request is removed, we also need to recalculate this level. At that moment only first in first serve method is selected without any priority.

- *Best combination of agents to provide list of services to multi users:* However in many cases, these services are distributed services and many agents may need to explore this services. In that case the above method is not suitable. We should show the best combination of agent. In that case service priority is given, a contract net protocol is worked to make the best combination of services. The bid is done by the slave agent in the system however the level of availability will be taken by using data mining from VDF local database. At that moment only first in first serve method is selected (No user priority).

At implementation time, the real available agents will be work together, regardless of the plan we have we will start a service when all the necessary agents are available.

To explore the services in this framework we propose the use of FIPA contract net protocol as a tool to evaluate the possible combination of partners in a service. This method explore the probability information in the virtual directory facilitator to calculate the available service so multiple users can access the same system and explore the services effectively.

3. EXPERIMENTS

The experiment validates the human-robot interface using slave agents, VDF and user agents. To do the experiment we used a home services system named ISSAC [6]. ISSAC, which stands for Intelligent, Sweeping Security Assistant companion, is a home mobile robot research sponsored by a Korean venture company, Woori-gisul.

Table 1. Service experiment result.

Mobile	2 Wheels (2BLDC Motors, Encoder) + Caster, 75 cm/sec	
Head-Eye	Pan & Tilt (2 BLDC Motors)	
Cleaner	1 Vacuum cleaner	
Appliances control	Remote IrDA Interface	
Sensors	Camera	1 Color USB Camera (Visual Information)
	Ultrasound	12 Ultrasound Sensors (Range finding, Obstacle detection)
	Infra-red	10 Sensors (Obstacle detection)
	Sound	1 Microphone+2 Speakers
Remote Control	Remote IrDA Interface	
Battery	Capacity	24V 12~20A, LY-Polymer
	Working time	3-4.5 hour without Vacuuming
Size and weight	H:800mm, D:400mm, W:20Kg	

To do this experiment, two service interfaces on ISSAC, cleaning service and behavioral customization service, are selected. The input for the cleaning service is time and zone to

be cleaned. When the conventional control system is used, ISSAC should have a permanent wireless connection to maintain the high available level. Because it is difficult to satisfy this, the service shows the useful point of the multi-agent system in sense of increasing the availability of the system. The behavioral customization service illustrates the other application of the framework. By assigning option to a group of slave agents in VDF, the behavior of all the robots with the same types can be changed. If there are some ISSAC robots in the same environment the behavior of the whole group can be changed by using only one service interface. When the option is changed on the main container, every robots placed in this environment will change its behavior and the robot becomes an environment awareness robot. Whenever ISSAC reaches the system again, its behavior will change accordingly.

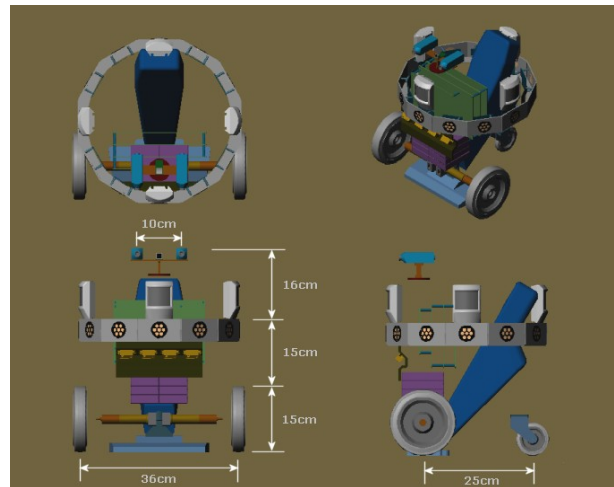


Fig 4. ISSAC Robot Model

We implemented a client/server model system to compare the usage of the two systems. The conventional system includes a server part that locates on the robot and a client part that can be downloaded through the Internet from the robot or work as the standalone application on user computer. The client/server can deal with offline situation by storing the users' request locally on the robot. However, users and robots are required to be online at the same time to send the request. The detail implementation of this system can be found in [7].

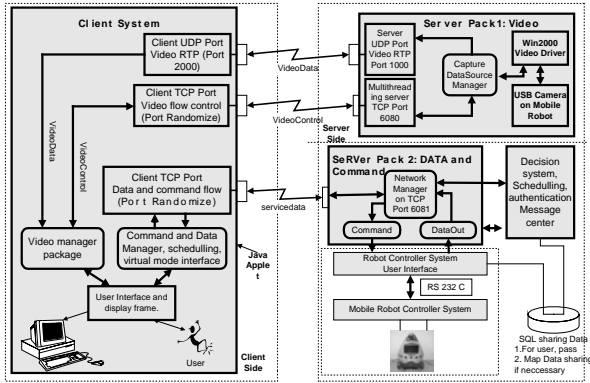
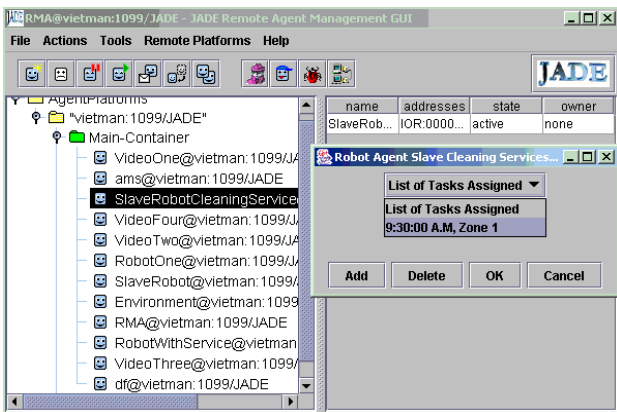


Fig 5. Conventional control System on ISSAC

As the framework is not only for ISSAC but also for any robot, a common offline control system such as a store and forward type system could not create the interface of the service because the features and behavior lists of each robot are different. Thus, the service interface should be designed on the ISSAC and then migrate to the AP later. The experiment is conducted with the UA on a PDA, iPAQ 5450, using JadeLeap [8]. This iPAQ connects to the network via an 11 Mbps wireless connection. The UA is written as an agent in an agent container on JadeLeap platform. In contrast to the conventional robotic control system, this UA does not have any particular information about robot or environment. After registering to the AP, the UA can search for the existing VDF, query the VDF on available services. VDF contains the services of robotic agents that are currently online and offline. After getting the list of services available in the system, UA may send query VDF to request a special service such as cleaning service. This service interface is cloned from the main container and migrated to locate on the user container. Then users run the interface on the user PDA, work with the service interface and leave the system.



6.a. Cleaning service, migrated to main container



6.b. Behavior customization services

Fig 6. Service Interface

Figure 6 shows the result of implementing our framework for cleaning service and behavior customization service. Both the proposed framework and conventional framework can offer cleaning services and customization services. However, with cleaning service, the proposed framework gives the ability of working offline with the services, which users cannot do on the conventional system. The behavior customization system shows the ability of changing the behavior of a group of robots. Moreover, the interface for changing this behavior is provided by the robots themselves, so many types of robots can provide this service if these robots are designed according to the architecture without changing the code at the VDF and UA. Each service of the proposed system is cloned and moved to user container separately on request. On the contrast, all the interface code of the conventional system should be loaded when users connect with ISSAC.

Table 2. Service experiment result.

	Average service starting time	Average time to complete a service	Time that both user & robot require to be online
Behavior Service Slave Agent 18.6 Kb	18 (second)	60 (second)	0 (second)
Cleaning Service Slave agent 28 Kb	20 (second)	120 (second)	0 (second)
Behavior Service on conventional system 96Kb	85 (second)	60 (second)	145 (second)
Cleaning Service on conventional system 96Kb	85 (second)	120 (second)	205 (second)

In table 2, the size for each service in the proposed system is smaller than the size of the service in the conventional system because the service interfaces in the proposed system

are separated from each other. Meanwhile, users should download the whole applet with all services in the conventional system. The time for starting any service in the conventional system is the same as the time for downloading the whole client program. The time to complete a service seems to be similar. However, the conventional system requires users and robots to be online at the same time.

4. CONCLUSION

The network HRI solution using a responsive environment and service management is good for autonomous robot working in dynamic environments with dynamic assigning tasks. Using this system, users can access the services at any time select a service and assign the task to the system. To explore these environments and working with robots, the use of VDF, slave agents and EMA is proposed. Through experiments, the proposed framework shows some advantages to the conventional tele-robotic control system. Users can access services of the system without installing the specific program for each robot. Total time for agent migration and synchronization is small enough to give the system a high ability level and help users save a lot of time to control the system. This method also reduces the time that robots need to be connected to the network and can tolerate some levels of connection disruption. Using the framework, a list of services is organized automatically and user can select the service to you without considering with robot will do the task. In the next period we plan to develop a service description language that can coordinate some robot in a complicated tasks and modify the VDF so that is can compile the task description script and distribute the task allocation to the partners in the system.

REFERENCES

- [1] Hideaki Takeda, Nobuhide Kobayashi, Yoshiyuki Matsubara and Toyoaki Nishida, "Towards ubiquitous human-robot interaction", in Working Notes for IJCAI-97 Workshop on Intelligent Multimodal Systems, pages 1-8, 1997.
- [2] Maxim Makatchev and S. K. Tso, "Human-Robot Interface Using Agents Communicating in an XML-Based Markup Language", in Proceedings of the 2000 IEEE International Workshop on Robot and Human Interactive Communication, Osaka, Japan, September 27-29, 2000, pages 270-275.
- [3] FIPA – Foundation for Intelligent Physical Agents <http://www.fipa.org>.
- [4] K. Huang and M. Trivedi, "Video arrays for real-time tracking of person, head, and face in an intelligent room" in Machine Vision and Applications, vol. 14, June 2003, pages 103-111.
- [5] Ali, K. and Arkin, R.C., "Multi-agent Teleautonomous Behavioral Control", in Machine Intelligence and Robotic Control, Vol. 1, No. 2, 2000, pages 3-10.
- [6] Bum-Jae You, Myung Hwangbo, Sung-On Lee, Sang-Rok Oh, Young Do Kwon, San Lim, "Development of a home service Robot 'Issac'", in Proceeding of the International Conference on Intelligent Robots and System (IROS2003), volume 3, pages 2630-2635.
- [7] Dong To Nguyen, Sang-Rok Oh, Bum Jae You, Myung Hwang Bo and Brian Kwang-Ho Lee, "A Control architecture for an Internet-based robot system", in Proceeding of the International Conference on control,

- automation and system (ICCAS 2002), pages 1677-1683.
- [8] F. Bellifemine, G. Caire, A. Poggi and G. Rimassa, "JADE - A White Paper", in Journal "EXP - in search of innovation", September 2003, volume 3, pages 6-19.