Proceedings of the 16th International Conference on Manufacturing Systems – ICMaS
ISSN 1842-3183

University POLITEHNICA of Bucharest, Machine and Manufacturing Systems Department
Bucharest, Romania

USING AGENT TECHNOLOGY TO DESIGN A COOPERATIVE SYSTEM

Cicerone Laurentiu POPA, Lidia Florentina PARPALA, Radu Constantin PARPALA

Abstract: *A cooperative system represents a system where different users or agents work together at the same project from different locations. The cooperative systems came from the distributed applications family and they are distinguish because the actors from the system work together to realize the same purpose. Agent technology has been recognized as an important approach for developing distributed intelligent design and manufacturing systems. They represent a new way of analyzing, designing, and implementing complex software systems. In this paper, we will present the main steps and the main elements needed for designing a cooperative system, and we will describe the main types of agents that we can use in the implementation of distributed systems.*

Key words: *cooperative system, agent technology, virtual enterprises, autonomous agent, communication.*

1. INTRODUCTION

In today economy context, concurrency, specialization and product diversification enterprises must respond promptly to the market demands to sustain their activities. In these conditions the enterprises consider collaboration with others to realize the product development in short time, with a lower cost, a short delivery time and a product that satisfy the client demand. Actual tend is to realize multidisciplinary cooperative systems for collaboration in real time between all professions involved in product development at the enterprise level. This collaborative environment actually constitutes a virtual enterprise [2].

Distributed and collaborative systems represent a new paradigm, and constitute a new interdisciplinary field between economic, computer, cybernetics, psychological and management field.

From the activity cooperative time perspective we have:

- real time communication;
- asynchronous time communication.

From the users place perspective communication can be from:

- single locations (face to face) - user are in the same place;
- different locations (distributed) - users cooperates from different geographic locations.

Agents that work in a cooperative system have access to common files, can exchange information (Fig. 1) and verify the project evolution [3].

2. COOPERATIVE SYSTEM DESIGN

The design of a cooperative system (Fig. 2) should start from:

- activities type;
- the number of the partners involved in the project;
- a better knowledge about de technologies needed in the system and their implementation;
- the infrastructure and the relation with the outside the system partners;
- what techniques and methods will be used in the system;
- cost of design, realization, maintenance and development of the system;
- lifecycle time of the product.

Next elements must be accomplished for the technologies and product integration:

- to realize a good communication and synchronization;

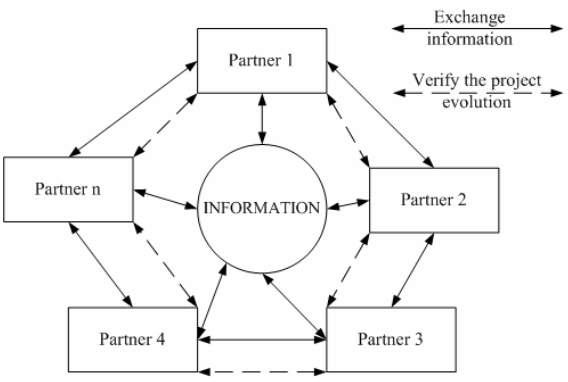

Fig. 1. Exchange the information in the cooperative system.

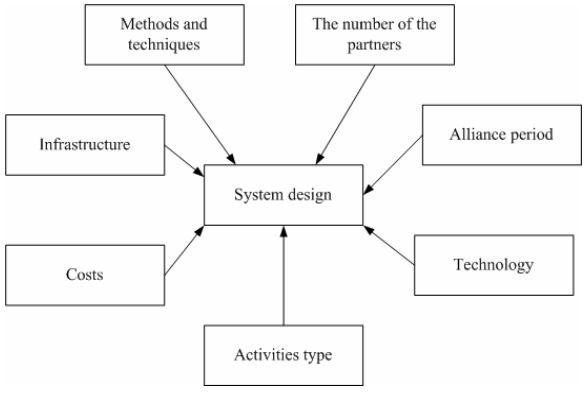

Fig. 2. Cooperative system design.

Fig. 3. Essential question put by organizations before they join to a virtual enterprise.

- opening to new technologies and methods;
- previous verification of potential risks;
- to be able to realize a lower cost and to reduce time;
- focused to client's demands;
- methods, information and instruments integration;
- systems able to support collaborative environments;
- education and personal training have an important role;

Some questions must be put before a company will join to a virtual enterprise based on cooperative system (Fig. 3):

Why? - a final purpose must be specified, a final product focused to client's demands.

What? - what products and technologies should be involved in the process.

How? - the organization strategies, the methods used for the final purpose realization .

Who? - from the beginning all the future partners and their responsibilities in the project should be known.

When? - when the alliance and main activities start and end.

Where? - the location of each actor from the project and the interaction methods with the group.

All the partners from the virtual enterprise should answer to the following questions:

- What methods should we use to realize new and innovative product?
- How the workflow can be accelerated?
- How can we make a better product with a lower cost?
- How can be sure we will be first on the market with this product?
- How can we prevent the delays in fabrication?

3. AGENT TECHNOLOGY

Agent technology involves building complex applications from autonomous, interacting components. It is particularly suitable for modelling, large-scale distributed simulations, analyzing complex systems, and improving and optimizing their behaviour. It can be perceived as a modelling paradigm, a problem-solving paradigm, or a software-engineering paradigm.

In agent-based programming, the agent is the basic element of distribution. Each agent serves as an independent component with its own local state and execution model. The agent designer can choose to assign a particular set of functionalities to an agent, specify the types of events and messages the agent may invoke or respond to, and implement those triggers and/or responses. In building an agent-based system, each agent behaves independently, interacts with other agents through the events and messages that are communicated from one agent to another.[8]

People involved in agent research have offered a variety of definitions, each hoping to explain the use of the agent. They define the agent from their point of view and from their domain perspective.

The AIMA Agent: "an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors."[6] This definition depend what the environment, the perception and the action mean. If the environment just assures the in-put and the out-put of an agent we can say that any program can be an agent, and this is not true.

The MuBot Agent:" the term agent is used to represent two orthogonal concepts. The first is the agent's ability for autonomous execution. The second is the agent's ability to perform domain oriented reasoning." This definition comes with an important property of agents: autonomy, and also talk about the main purpose that the agent must realize, but the author doesn't mention anything about the environment.

The KidSim Agent: "we define an agent as a persistent software entity dedicated to a specific purpose. Persistent distinguishes agents from subroutines; agents have their own ideas about how to accomplish tasks, their own agendas. Special purpose distinguishes them from entire multifunction applications; agents are typically much smaller." In this definition the authors try to explain the differences between an agent and a usual program. Also they insist on the purpose approach when they define the agent.

We can say that the agent is a computer program that acts autonomously on behalf of a person or organization. Each agent has its own thread of execution so that it can perform tasks on its own initiative. When an agent travels, it transports its state and code with it. In this context, the agent state can be either its execution state, or the agent attribute values that help it determine what to do when it resumes execution at its destination.

4. AGENT'S CHARACTERISTICS

Agents have also other characteristics [9]:

- autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
- pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative;
- social abilities: agents interact with other agents (and possibly humans) via some kind of agent-communication language;

- mobility: the agent is able to transport itself from one machine to another;
- coordination;
- communication;
- cooperation.

Intelligent agents must have:

- knowledge: "Agent A knows that his environment is instable";
- convictions: "Agent A decide to do action x because he knows that this is the right way for the purpose realization;
- purpose: "Agent A wish to obtain y information";
- intention: "Agent A wish to move on the server B";
- choices: "Agent A choose to move on server B and not on server C";
- commitment: "Agent A continues to do action x until he will realize the purpose given by the user".

5. AGENT'S OBJECTIVES

Figure 4 presents the main steps that an agent must do to realize the purpose gave by the user. First the agent must recognize this new objective received from the user. Also he will search for information in his environment and will interrogate the user for additionally date. When he has all the information needed he will evaluate the opportunities of purpose realization and will decide to follow the right alternative.

In this process the agent will negotiate with other agents from his environment because these agents can contribute to the main purpose realization or perhaps they have the same objective.

After accomplishing these steps the agent must take the final decision: which alternative is the best, faster and cheaper. After the purpose completion, the agent generates a report to the user with all the information from all the steps that he passed.

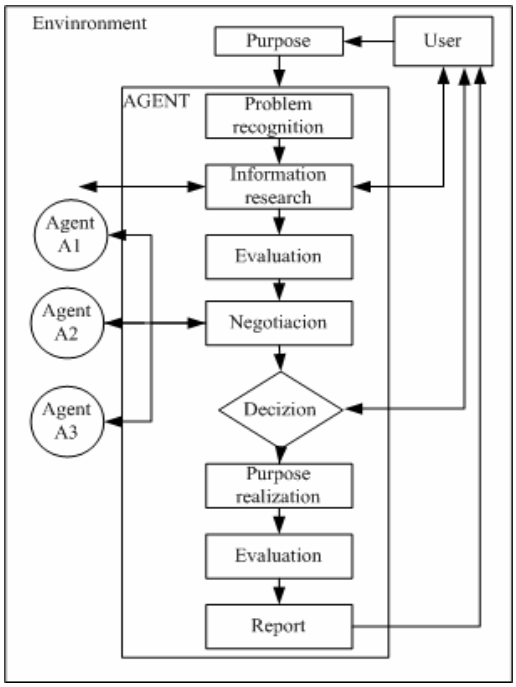

Fig. 4. Agent objectives.

If the objective is successful realized he will report the way of doing this, the cost and time. If he failed in the object realization, the agent must specify the cause of failure.

6. AGENTS TYPES USED IN COOPERATIVE SYSTEMS

When we decide to use agent technology for a cooperative system implementation first we must study the main types of agents that are necessary in the system. Next we will present some types of agents that we can use in a common distributed system.

The user agent main task is to guide the user to use the system at his best performances. For example a costumer make a request in the system, this agent will help the user to make all necessary steps to accomplish the request.

Interface agents main task is to adapt the information from the application in a format that can be understood by the user or his personal agent. They are autonomous and utilize learning to perform tasks for their users. This class of agent is also used to implement assistants, guides, memory aids, and filters; perform matchmaking and referrals; or buy and sell on behalf of the user.

In a design system agents must assure the communication, cooperation and to improve coordination for the actors involved in the system. They have to allow to all actors to change information and because in such systems are involved actors from different domains, they need to be able to take decisions, to cooperate and to negotiate the relations between these actors.

An agent that will realize a software version check is useful in the system. He must be able to offer some solutions to prevent the appearance of errors, losing information and he must keep records about all the activities from the system.

For example agent A ask to process a part on M1 machine, if our agent will have all the information he can tell to agent A that the part will be process faster and chipper on M2 machine. Or an error example: agent A asks to acquire a tool-machine, but the task is impossible to realize because the price is too high for this department budged and our agent have to prevent that. To prevent the errors is an important task because these errors can stop all the process and will delay the delivery time.

In the system we must have agents for machine and robots control. These agents have all machines or robots information, the types of parts that can be machined, the process time and cost, the material and tools needed. Also they will know a tool is damaged and need to be replaced.

For example, when a problem needs to be solved the agent will follow a totem-pole. First the agent will check the general condition of machine. He will check if the agent which controls the energy system had cut the energy to our equipment and if is true will ask the reason. He also will check the tool if is damaged or worn and need to be replaced. In this case he will send a note to the agent that manages the store department to deliver a new one. Also he will gather information about the causes of damage.

An agent that manages the delivery system is needed. He will have all the information about the products, delivery times and costumers. He must be able to check on the Internet: the roads condition, the weather, traffic and he will decide which way is the best for delivery.

Also an agent that will manage all the agents that have responsibilities in the manufacturing department is useful. For example he can decide to replace equipment E1 with a new one E2 to continue the production in good condition. Some problems can appear. First he will check if the new equipment can be placed in the department (maybe is too big), he can decide to put it between E3 and E4 for the best process optimization etc.

Many other types of agents can be involved in the system. For example: a security agent that manages the inputs and the outputs of other agents in the system, coordination agents, auxiliary agents etc. [5].

7. THE ADVANTAGES OF COOPERATIVE SYSTEMS

The advantages of using cooperative systems are:

- a good intellectual property protection, because the organization gives to the partners just the information needed
- a lower cost for infrastructure adaptation to project and costumers demands;
- a good flexibility;
- better relation inside the organization;
- access to the date base is made by security levels according to the partners importance, his implication in the project and the type of information demanded;
- gives a workflow automation level proper to the company demands;
- gives access to all web services, and also for the teams that work in a distributed environment;
- must be able to keep the standards for operation systems and for information security;
- allow real-time interaction and visualization for the partners that realize a lower cost and reduce time for the product development;
- gives the same interface to all actors involved that allow communication, organization and collaboration.

8. DOMAINS AND APPLICATION

Agent technologies are suitable for domains where:

- highly complex problems need to be solved or highly complex systems to be controlled;
- the information required for solving problems or controlling systems is distributed and is not available centrally, because:
- it is distributed geographically;
- the agents do not want to share it;
- in domains with dynamically changing environment and problem specification;
- different pieces of software need to be integrated in a run time and in the design times.

Possible application:

- manufacturing: planning highly complex production, control of dynamic, unpredictable and unstable processes, diagnostics, repair, reconfiguration and replanning;
- virtual enterprises: forming business alliances, forming long-term/short-term deals, managing supply chains;
- internet agents: mainly for intelligent shopping and auctioning, information retrieval and searching, remote access to information and remote system control [7].

9. CONCLUSIONS

Agent technology represents a new way of analyzing, designing, and implementing complex software systems. Using this information we intend to realize a distributed system architecture supporting the exchange of information and the sharing of knowledge between actors, in order to study the agent interaction and analyze their behaviour in this system.

In the future we intend to realize a cooperative system architecture supporting the exchange of information and first we will focus on agents for manufacturing department (agents for machine and robots control).

REFERENCES

[1] Caglayan, A.; Harrison, C. (2002). *Agent Sourcebook: Commercial Agent Software,* Wiley Computer Publishing

[2] Drăgoi, G. (2003). *Întreprinderea integrată: metode, modele, tehnici şi instrumente de dezvoltare şi realizare a produselor* (*Integrated enterprise: methods, models, techniques and instruments of developing and achieving products*), Edit. Politehnica Press, ISBN 973-8449-25-1, Bucharest.

[3] Farley, J. (1998). *Java™ Distributed Computing*, ISBN 1-56592-206-9E.

[4] Orlikowski, W.; Yates, J. (2002). *Structuring Interaction through Communicative Norms*. Journal of Business Communication, pp. 13-35.

[5] Popa, C.L. (2005). *An Agent Classification From the Environment Perspective,* Proceedings of 16th International DAAAM Symposium, Katalinic, B. (Ed.), pp. 309-310, ISSN 1726-9679, Opatija-Croatia, Published by DAAAM International, Vienna

[6] Russell, S. J., Stuart, J., Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, NJ, Prentice Hall.

[7] Vokrinek, J., Rollo, M., *Agent System Design*. Available from: `http://agents.felk.cvut.cz/` *Accessed*: 2005-05-10.

[8] Weiss, G. (200). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, ISBN 0-26223-203-6, Cambridge, Massachusetts.

[9] Wooldrige, M. (2002). *An Introduction to MultiAgent Systems*, John Wiley&Sons.

Authors:

Eng, Cicerone Laurentiu POPA, PhD student, University POLITEHNICA of Bucharest, Machine and Manufacturing Systems Department,
E-mail: `Laurentiu.Popa@mix.mmi.pub.to`
Eng, Lidia Florentina PARPALA, PhD student, University POLITEHNICA of Bucharest, Machine and Manufacturing Systems Department,
E-mail: `Lidia_Parpala@yahoo.com`
Eng, Radu Constantin PARPALA, PhD student, University POLITEHNICA of Bucharest, Machine and Manufacturing Systems Department,
E-mail: `pmc@mix.mmi.pub.ro`