An Overview of the Use of Mobile Agents in Virtual Environments

Gonzalo Méndez[1], Pedro Pérez[2], Angélica de Antonio[3]

Facultad de Informática, Universidad Politécnica de Madrid
Madrid, 28660, España
[1]gonzalo@gordini.ls.fi.upm.es, [2]pperez@ts.es,
[3]angelica@fi.upm.es

Abstract. In the field of agents, there are a wide variety of them, such as learning agents, planning agents or communicative agents. One of the youngest members in the family are mobile agents, which provide us with the interesting feature of mobility in order to perform their tasks in different machines. In this paper, we will see some of the current uses of mobile agents and we will suggest how we could use these agents along with Virtual Environments in order to enhance them and open a new world of possibilities for the users of these applications.

Introduction

With the development of computer science, modern society is suffering important and drastic changes in fields as important as education, medicine and business in general. However, there are fields, such as entertainment, where these changes are traditionally better accepted and considered. Some of the technologies that are having better acceptance are Internet, three dimensional applications (Doom-like games), and, little by little, personal assistants that help the user carry out some common, repetitive tasks. These technologies lead us to more serious, practical and useful applications, such as Distributed Systems, Virtual Environments (VEs) and Intelligent Agents. Among them, the ones that are having a faster development are VEs, although they are taking advantage of all the advances that are taking place in the other two fields, thus giving birth to Distributed Virtual Environments (DVEs) and Intelligent Agents that perform their activities inside the DVEs. Besides, agents are also taking advantage of distributed computing, and there is currently a great deal of work going on in the field of mobile agents [1]. However, there is still a field where there seems to be little activity: the integration of both kinds of agents, intelligent and mobile, in an environment as potentially powerful as VEs. This is where our current research is aimed at.

Mobile Agents

The use of Internet has increased during the last years and it will not stop in a foreseeable future. Among the different possible uses of this global public network, we can find an essential base: sharing information [17]. The most common way to carry out the searching tasks is by ourselves, but why not assigning this task to a program instead of to ourselves? We can find the answer to this question in *mobile agents*, a concept that emerged in 1994 with the development of *Telescript* [15] and that has experienced an important support with the development of *Aglets* [18], MASIF [29], Mole [19], [20], [21] or Concordia [22], [23].

Mobile Agents are small pieces of software, with intelligence and autonomy, that can travel along the network from one host to another in order to execute their code and to retrieve some information. This information will be sent back to us when the mobile agent has finished its work. Therefore, we ought to have our own agents in these systems and host programs, that have to allow the execution in the host node. We have to be careful with security in the host node because it has to execute code made by third parties. We must establish a fixed set of available requests and a separate work space from the host itself.

In fact, this sort of agents are software that represent one client and have the ability of to go through the network while they take decisions in order to attend a service.

One agent should have three dimensions from representation and mobility to intelligence:

- Representation: One agent represents one users to other systems or agents.
- Mobility: One agent is moved through the network and can collect information while they travel from one machine to another.
- Intelligence: Ability to apply knowledge in order to solve problems and to take decisions with the data they collected.

Other good characteristics in agents are autonomy, flexibility, security [7], [16], [25] and ability to cooperate with other agents [4], [6].

These agents are programmed under some parameters, and some intelligence is previously included in them. For instance, we can use them to perform tasks in a remote system to which you can send these pieces of software and obtain the results later, instead of keeping the communication along the whole process [30]. The reasons for this way of working are not only economical, but simply practical: we can pay attention to other tasks while the agent is carrying out its (our) tasks. One example of this kind of system is the control of robots when exploring the surface of a planet. We could have to carry out some task in real-time and the delay in long distance communications, many times due to the narrow available bandwidth[7], [17], may be unacceptable[3], [14]. In this case, the use of mobile agents would solve the problem [5]. In our case, we can create mobile agents in order to make them look for VEs that may be interesting for us and take our place as a virtual representative agent that can negotiate with other people or agents.

Techniques

Internet is the natural environment for mobile agents, but it is not the unique world. These pieces of software collaborate in the network through a client-server model with distributed computation. We can designed this distribution of processes with different programming techniques, such as Remote Procedure Call (RPC), Remote Method Invocation (RMI, with Java), Common Object Request Broker Architecture (CORBA) in order to have communication with different and separated processes and objects. The aim of these different programming techniques would be to have mobility, flexibility and concurrence.

Mobile agents involve several technologies, from communication systems to knowledge systems. One strategy is the use of CORBA for communications as a general framework. It is a standard, widespread communication method with the interesting features of being multiplatform and available for different programming languages, something almost essential for an agent that needs to travel along the network and stay in several systems. Besides, CORBA fits well because of its object oriented nature. The mobile agent can be fit in an object structure that can be easily implemented in CORBA [31]. In the case of knowledge systems we would have different approaches, ranging from rule-based systems to fuzzy systems.

In theory, mobile agents don't need to learn from the environment, but, in the case of VEs, this should be a really recommendable feature.

Mobile Agents Inside Virtual Environments

Once we have seen what mobile agents are and how they work, it is time to see how they can be useful inside a VE. We must take into consideration that there are very different kinds of VEs: educational, medical, social, etc. Thus, they are not likely to have the same utility in all of them, and it may not be a good idea to use agents in all of them. Besides, the VE needs to have some special features in order to let mobile agents perform their activities in a consistent, secure way. The same applies to Intelligent agents, since they must have an interface with which mobile agents can interact in order to give instructions and receive results.

Possible Uses of Mobile Agents in Virtual Environments

Asking why a mobile agent is useful in a VE is like asking why an agent is useful in a VE. Either because they provide us with useful information or because the interaction with them has a positive effect for us, the fact is that agents are getting more and more importance inside VEs.

However, in many occasions, an agent inside a VE is just that: 'only' an agent. But if we consider an agent as a representative of a person, then the utility of mobile agents starts to be clearer. As such representative, this agent must accept some high level instructions from the user, and then it will decide which is the best way to carry out the specific action [2].

So now the question is different, and we should try to figure out what kinds of actions a mobile agent can perform that could be interesting for us. The generic uses of mobile agents are, basically, two: gathering information and locating other agents. Then, these two objectives are perfectly suitable for VEs.

First, it may be quite interesting to locate the representative of another person in order to let the user talk to that person. We can send our agent to explore a series of VEs until it locates the person we are looking for. It can then come back to us with the information and we can connect to that particular VE in order to meet the person we are looking for.

Second, it may also be interesting to send our agent to explore some unknown VEs in order to know their purpose and see if any of them may have some interest for us. What's more, once we have decided that one of those VEs is interesting, we could send our agent back in order to perform some action in it instead of having to do it by ourselves. For example, it might queue to get some tickets to attend to an event where the number of connections is limited, until we can connect to it and see it, or it could be sent to an auction with instructions to bid for some of the objects on sale.

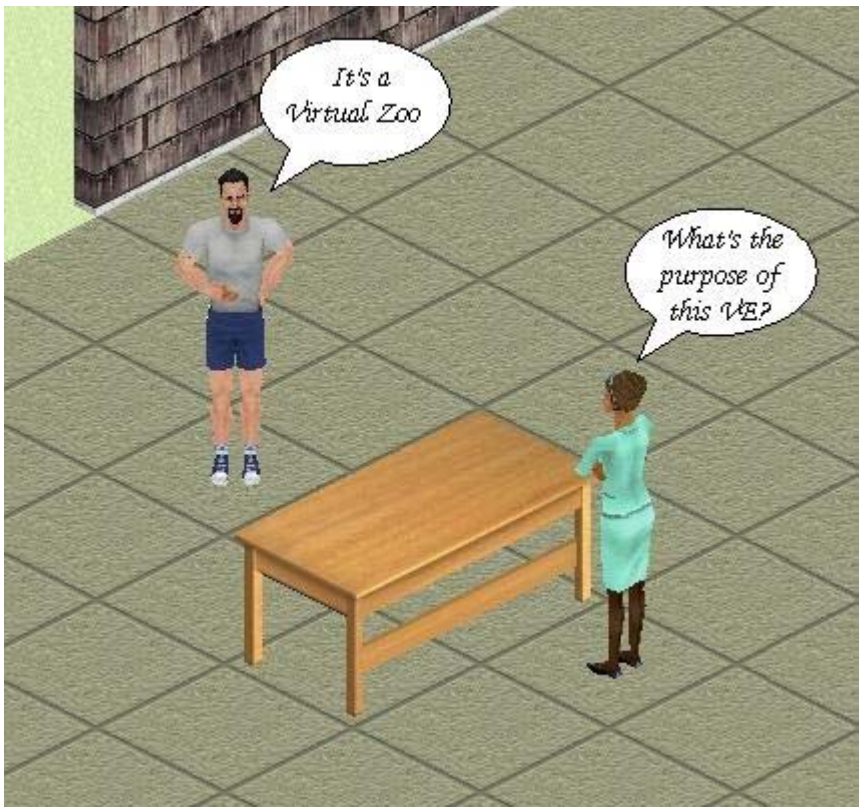

Fig. 1. A Mobile Agent at the reception of a VE

Features of the Virtual Environment in order to Accept Mobile Agents

Once we have seen why a mobile agent could be useful working inside a VE, there are two things we must do in order to make sure that the mobile agent will be able to move and interact with other agents and objects.

First of all, we must define the characteristics that the VE must have in order to allow the presence of mobile agents. The idea is for the VE to behave as a normal agency for mobile agents. Since agents from different sources will be able to reach our VE and we want VEs of very different natures to be accessible by these agents, the VE must have:

- Security: visiting agents must be able to move around the VE, but they must not be able to perform certain actions that may be risky either for the VE or for other visitors.
- Standard Interface: due to the different natures of agents and VEs, a standard interface must be defined in order to let the mobile agent ask the VE for information that may be useful for the user who has sent the agent. This information may be the purpose of the VE, users and agents connected to it, ongoing and future events that will take place in the VE, etc.

For the second feature to be possible, there must be a reception at the entrance of the VE where the mobile agent may ask for all this information. As we will see a bit later, it won't be necessary for the VE to have additional features other than the ones we have already defined, since once the mobile agent has decided to enter the VE, it won't interact directly with the VE, but by means of another agent. This will also be an advantage in terms of security, since the mobile agent will carry out only the actions that the other agent allows it to do.

Features of an Intelligent Agent in order to Be Controlled by a Mobile Agent

The second task that must be carried out is the definition of a standard interface that allows the communication between mobile agents and the agents that inhabit the VE, which will be used by the mobile agent in order to visit the VE and interact with other agents and objects.

This is the most complicated part of the problem since we are trying to:

- Communicate different kinds of agents inside VEs with a very different purpose. For this to be possible, the communication must be general enough in order to grant the access to the VE to any agent. If not, we will be limiting the access of certain kinds of agents to certain VEs, since not all the agents will recognize the purpose of all kinds of VEs (i.e. an agent designed to visit museums won't be able to visit a spacecraft). Besides, building agents that are able to move in very different VEs taking advantage of all their features would imply that their size would be so big that sending them through a network would take a considerable amount of time.

- Do something useful for the user. This is actually the main purpose of the whole application, since there is no point in sending agents through a network if they are not going to perform any beneficial action (preferably for the agent's owner, but also for other users).

Therefore, the problem is to design a communication interface general enough so that the access to the VE is not restricted for any kind of agent and specific enough so that they can perform actions that will result in a real benefit for their owner.

The agent that inhabits de VE must offer:

- The possibility to receive high level, general commands from the mobile agent. These orders will allow it to visit the VE and interact with the rest of the users and agents, so they must be actions such as 'Move', 'Ask', 'Look', 'Take', 'Look For' and the like. These actions must be provided by all the agents inhabiting all the VEs that can be visited by our mobile agent.

Fig. 2. A Mobile Agent instructing its proxy Agent

- The possibility to receive VE-specific commands. For example, our mobile agent should be able to pay a ticket to enter a museum, to bid in an auction to buy some of the products on sale, to ride a horse in a tour around the Virtual Zoo or to manipulate complex machinery in a Nuclear Power Plant. The most complicated

part of this task is that the mobile agent must be able to command these actions without knowing the purpose of the VE beforehand. This objective could be achieved through the use of ontologies. The agent might enter the VE and load a specific module, already present in the VE, which would allow the agent to perform the VE-specific tasks. The use of ontologies [32] would allow the agent to know the effect of these actions, and it could decide whether it would be interesting to perform them or not.

How the Whole System Works

There are two basic functionalities that our system should have; the main idea is for them to be used together, but they will be described separately in order to allow the implementation of any of them independently.

The first objective of the system would be the exploration of different VEs in order to figure out what is their purpose and what other agents or people are inside it at that moment.

The main goal is to find VEs that may have a special interest for the user, along with people with whom the user would be interested to get in touch.

This is a problem that mobile agent based systems have partially solved long ago. You can already make your agent travel to other machines and ask for some other agent in order to perform some activity with its help. Thus, the basic idea is to extend this functionality so that it provides more thorough information about the VE and its inhabitants.

The user would send the agent to a remote host with detailed instructions [13]: to look for VEs with certain features or some specific users or agents in it. The agent will travel first to that remote host, which will have a reception where the agent will ask some questions, but without entering the VE, this is, without having a real presence and a physical appearance, since we are not interested in whatever is inside it, yet.

The agent will be provided with three pieces of information: the general purpose of the VE, the number and nature of the users currently connected to it, and a list of known VEs with similar characteristics, in order to let the agent keep on with its search. The agent will provide the VE with the list of VEs it has already visited, in order to let the VE refresh its list of similar places with some possibly new ones. Besides, a list of the next events that will take place in the VE could be asked for, so the user will be able to decide whether he will try to attend to any of them.

With this new list at hand, the agent will travel from one VE to another until all of them have been visited, some predefined time has expired or the user we were looking for has been found; at that moment, the agent will go back to its originating machine in order to provide the user with all the information that it has gathered in that trip.

Once the user has received these results, two different courses of actions might be followed [13]: the user may decide to do things himself or he may want the agent to perform some other action.

In the second case, he may want the agent to get in touch with some of the agents or avatars that inhabit one of the VEs or he may want the agent to attend to one of the events that will take place in another VE.

The agent will then travel to the VE, register in the reception, and enter the VE. At this moment, a new VE specific agent will be assigned to the mobile agent in order to 'guide' and represent the mobile agent inside the VE. Now, the mobile agent will adopt a physical appearance in the VE, and other agents and users will be able to interact with it as if it were a 'normal' user.

Since the mobile agent is in touch with one of the intelligent agents that have been made to populate that VE, the mobile agent can give it the instructions necessary to achieve the goals planned by the user.

If we are looking for some other agent or avatar, it will be easy to tell the guiding agent to look for it. Once it has been found, the mobile agent will ask what the user has told it to find out or it will let the user now that the target has been located in order to let him make the appropriate decisions.

We may also want our agent to perform some actions in the VE for which the instructions must be domain dependent, this is, if we want our agent to bid in an auction or to buy shares in the Virtual Stock Market, it must be told exactly what to do. For example, we might want it to buy $4000 of the must profitable shares during the last month, and for that, specific instructions must be given. In other cases, not so specific instructions must be used in order to, for example, tell the agent to occupy our place in a virtual event where the number of attendants is limited or is so big that we want to get a good sit in order to see the show in detail.

But there are some actions that are so domain-specific that we cannot train our agent to perform them, since it would take so much time that it would be better for us to do them by ourselves. If these actions are critical, it would be better not to let the agent do them (and, in all probability, the remote host would not let the agent travel to it). But for not so critical actions, it would be a good idea to have some mechanism to tell the mobile agent what actions can be performed in that particular VE. For example, it might be possible to build a Virtual House where other agents might go to look for us. Since that feature would not be very common in other VEs, there is no point in including it as a standard feature for all the VEs that accept the presence of our mobile agent, but it should be possible for it to make use of this feature if the VE allows it.

One possibility to solve this problem is the use of ontologies. With them, our agent might arrive to the VE, learn what actions he may perform and what their effects would be. For the moment, this solution is only suitable for exploring VEs in order to see what can be done inside them and how. Then, the agent might inform its owner, who would decide whether he wanted to be a stable inhabitant of that VE or not.

The other possibility is for the agent to load a local module that will be in charge of letting the agent explore the specific actions that may be performed inside the VE and thus inform its owner of the different possibilities that exist in that particular VE.

Implementation

At the time of writing this paper, we are currently working on the implementation of the described functionalities in a prototype system, which is addressing interesting design issues [27], [28].

For the realization of the VEs, we are using WorldUp and Active Worlds, since both of them have the interesting feature of allowing us to call external routines programmed in the C++ programming language stored in dynamic libraries. In a close future, the next step will be to implement our own engines, since it is a more flexible solution and the integration with mobile agents will be simpler. For the moment, however, these two applications allow the quick development of a prototype with which to obtain the first results in our experiments.

Besides, there are several CORBA implementations available for C++, which will allow us to work out the communication between the different remote hosts and the mobile agents [31].

The basic idea is to integrate the agency with the VE, so when an agent arrives to the VE, it can immediately have a physical appearance and can be assigned a local agent to guide it and help it during its visit to the VE. It is also useful in order to implement the reception of the VE, since it will have access to the most actualised information about the current state of the VE: inhabitants, events, etc.

When the VE is loaded, the agency will also be created, with an empty list of inhabitants and events, and it will inform other hosts of its presence in order to let mobile agents know its existence.

Every time an agent arrives, it will supply information about the VEs it has already visited and will ask either for some information or for permission to enter the VE. In the first case, he will be provided all the information it asks for, as long as it is available. These requests have been thought to be implemented using KQML [9], [10], [11], [31], which provides a standard language to construct them, although KIF [8] and Ontolingua [12] are also being evaluated.

In the second one, once the agent is allowed to enter the VE (according to some security policies, such as precedence of the agent or number of users already connected to the VE), an agent will be created in the VE and it will be assigned a physical appearance. All the communication between the mobile agent and the VE and its inhabitants will take place using the local agent as a proxy, and the mobile agents will be informed of the ontologies to be used and necessary modules to be loaded in order to allow detailed communication with the local agent, so that specific actions can be carried out.

When the mobile agent leaves the VE, the local agent will be destroyed, and the agency will be informed that the agent is no longer in that host. The mobile agent will write down the necessary data about the VE in order to inform other VEs of its existence.

Conclusions

As we have mentioned before, although the system has not been implemented yet, we are currently working on it, and it is not very far in the future the moment when these existing technologies will be fully integrated and working in different kinds of VEs.

Of course, there is still a lot of work to do, and some years have to pass before a standard is defined in order to allow open communications between different kinds of systems and agents, but it is undeniable that the integration of mobile agents and

Virtual Environments will be really beneficial for users in general and researchers in particular, due to its potential as a source of information and as a way to share it with other people.

All the technology necessary to develop this kind of systems is currently available, so it is just a matter of time that we see these systems working and providing a useful functionality for their users.

References

1. S. Franklin, A. Graesser. ***"Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents"***. Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages. Springer-Velag. 1996.
2. B. Jung, J. T. Milde. "An Open Virtual Environment for Autonomous Agents Using VRML and Java". VRML 99. Paderborn, Germany. 1999.
3. S. Stone, M. Zyda, D. Brutzman, J. Falby. ***"Mobile Agents and Smart Networks for Distributed Simulations"***. Proceedings of 14th Distributed Simulations Conference, Orlando, FL, March 11-15, 1996.
4. Nick Jennings et al. ***"Autonomous Agents for Business Process Management".*** International Journal of Applied Artificial Intelligence, 2000.
5. Chavez, Guttman & Moukas. ***"Challenger: A Multi-agent System for Distributed Resource Allocation"***. Proceedings of the First International Conference on Autonomous Agents '97, Marina Del Ray, California, 1997.
6. Maes & Guttman. ***"Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce".*** Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98). Paris, France, July 3-8, 1998.
7. Maes, Guttman & Moukas. ***"Agent-mediated Electronic Commerce: A Survey".*** Knowledge Engineering Review, June 1998.
8. Knowledge Interchange Format (KIF). http://logic.stanford.edu/kif/
9. T. Finin, D. McKay, R. Fritzson, and R. McEntire. "*KQML: An Information and Knowledge Exchange Protocol*" Knowledge Building and Knowledge Sharing, Ohmsha and IOS Press, 1994.
10. T. Finin, Y. Labrou, and J. Mayfield. "*KQML as an Agent Communication Language*" Software Agents, MIT Press, 1997.
11. R. S. Cost, J. Lakhani, I. Soboroff, T. Finin, E. Miller, C. Nicholas. "TKQML: A Scripting Tool for Building Agents". Proceedings of the 1997 Conference on Agent Theories and Agent Languages (ATAL97), Newport, RI, Jul. 1997.
12. T. Gruber. "*Ontolingua: A Mechanism to Support Portable Ontologies*". Stanford University, Knowledge Systems Laboratory, Technical Report KSL-91-66, March 1992.
13. Maes & Chavez. ***"Kasbah: An Agent Marketplace for buying and Selling Goods".*** Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, April 1996.
14. Papaioannou. ***"Mobile Agents: Are They Useful for Establishing a Virtual Presence in Space?".*** Agents with Adjustable Autonomy Symposium, part of the AAAI 1999 Spring Symposium Series.
15. J. E. White, "Telescript technology: the foundation for the electronic marketplace", White Paper, General Magic, Inc. 1994.
16. R. Gray, "*Agent Tcl: A flexible and secure mobile agentsystem*", PhD thesis, Dept. of Comp Science, Dartmouth College, June 1997.

17. T. Papaioannou, J. Edwards. ***"Mobile Agent Technology Enabling The Virtual Enterprise: A Pattern for Database Query".*** Agent Based Manufacturing workshop, part of Autonomous Agents '98.
18. Clemments, T. Papaioannou, J. Edwards. ***"Aglets: Enabling the Virtual Enterprise".*** ME-SELA '97, p425.
19. M. Straer, J. Baumann, F. Hohl. "*Mole - A Java based mobile agent system*". ECOOP '96 Workshop on Mobile Object Systems, 1996.
20. http://mole.informatik.uni-stuttgart.de/
21. J. Baumann, F. Hohl, K. Rothermel, and M. Strasser. "*Mole - Concepts of a Mobile Agent System*". Technical report 1997/15. Fakultaet Informatik, University of Stuttgart, Aug. 1997.
22. A. Castillo, M. Kawaguchi, N. Paciorek, D. Wong. "*Concordia as Enabling Technology for* Cooperative *Information Gathering*". Japanese Society for Artificial Intelligence Conference, Tokyo, Japan, June 17-18, 1998.
23. http://www.merl.com/HSL/Projects/Concordia/
24. T. Papaioannou, J. Edwards. ***"Manufacturing Systems Integration and Agility: Can Mobile Agents Help?"*** Special Issue of Integrated Computer-Aided Engineering, IOPress, January 2001.
25. Papaioannou & Edwards. "*Using Mobile Agents To Improve the Alignment Between Manufacturing and its IT Support Systems*". Journal of Robotics and Autonomous Systems, Vol 27, pp45-57.
26. T. Papaioannou, J. Edwards. ***"Towards Understanding and Evaluating Mobile Code Systems".*** Proceedings of Cooperating Information Agents (CIA-2000).
27. Y. Aridor, M. Oshima, "*Infrastructure for mobile agents: requirements and design*". Proceedings of the 2nd Int. Workshop on Mobile Agents, 1998. Lecture Notes in Computer Science, Vol. 1477, pp. 38-49, Springer Verlag, Berlin.
28. T. Wojciechowski, P. Sewell. "*Nomadic Pict: Language and infrastructure design for* mobile *agents*". Proceedings of ASA/MA '99 (First International Symposium on Agent Systems and Applications / Third International Symposium on Mobile Agents), Palm Springs, CA, USA, October 1999.
29. D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, J. White. "*MASIF: The OMG Mobile Agent Sys-tem Interoperability Facility*". Proceedings of the 2nd International Workshop on Mobile Agents, LNCS 1477, pp. 50-67, 1998.
30. J. Baumann, F. Hohl, N. Radouniklis, K. Rothermel, M. Strafier. "*Communication* concepts *for mobile agent systems*". Proceedings of the 1st International Workshop on Mobile Agents (MA'97), 1997.
31. D. Benech, T. Desprats, Y. Raynaud. "*A KQML-CORBA based Architecture for Intelligent Agents Communication in Cooperative Service and Network Management*". IFIP/IEEE International Conference on Management of Multimedia Networks and Services, Montreal, Canada, July 8-10, 1997.
32. B. Chandrasekaran, J. R. Josephson, V. R. Benjamins. "*What are Ontologies, and Why Do We Need Them?*". IEEE Intelligent Systems and their applications, Vol. 14, Num. 1, Jan. 1999.