mIVA: Why to use Mobile Agents in Virtual Environments and Wireless Devices?

Pedro Pérez Rodríguez[1] and **Gonzalo Méndez Pozo**[1] and **Angélica de Antonio Jiménez**[1]

Abstract. This paper is focused on a special type of agent which provides us with the interesting feature of mobility in order to perform its tasks in different machines: mobile agents. We will show some of the current uses of mobile agents and we will suggest how we can use these agents along with Virtual Environments and Wireless Devices in order to open a new world of possibilities for the users of these applications.
Keywords. Mobile Agent, Intelligent Agent, Virtual Environment, Mobile Intelligent Virtual Agent.

1 INTRODUCTION

Internet is a really efficient communication environment when we are interconnecting dispersed systems. There are a lot of uses we can think of for this communication environment, from basic communication systems like file transfer and email services, to really sophisticated ones, among which we have educational and learning systems and three-dimensional applications like games and personal assistants. In all of them, it is possible to integrate Virtual Environments (VEs) in which we can develop our activity. VEs are three-dimensional environments that describe a virtual world where we can carry out different tasks, depending on their purpose, ranging from training in knee surgery to visiting a museum or just chatting. VEs are inhabited by avatars, a representation of ourselves, controlled by us, or by autonomous agents, and all of them can interact in order to provide some benefit for the inhabitants of the VE, may that benefit be some kind of information or just fun. However there may be certain tasks to be performed that may have no interest for the user, either because they are repetitive, boring or superficial, but, since they must be carried out, we might like to entrust an agent with these tasks. It is here where we can find the link between Virtual Environments and Agents and with Distributed Virtual Environments (DVEs): 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 a VE. This is where our current research is aimed at [2].

2 MOBILE AGENTS

One of the possible and more extended uses of Internet is sharing information [3]. The most common way to carry out the searching task is by using some of the existing search engines or even a software agent or assistant, but what if this information is not in a web page? And what if it changes very quickly or even if we can take part in that changing process? We can find the solution to this problems in mobile agents, a concept that emerged in 1994 with the development of Telescript [4] and that has experienced an important support with the development of Aglets [5], MASIF [6], Mole [7], [8], [9] or Concordia [10], [11].

Mobile Agents are small pieces of software, with intelligence and autonomy, that can travel through 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, there must be host programs that allow the execution of the travelling code 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 that may not be trusted, so a fixed set of available requests and a separate workspace from the host itself must be established.

A mobile agent should have three dimensions, from representation and mobility to intelligence:

- Representation: One agent represents one user in terms of interaction with other systems or agents.
- Mobility: An agent travels through the network and can collect information while it moves from one machine to another.
- Intelligence: Ability to use knowledge in order to solve problems and to make decisions taking into account the acquired data.

Other good characteristics in agents are autonomy, flexibility, security [12], [13], [14] and ability to cooperate with other agents, or even to compete against them [15], [16].

Mobile agents are programmed following some well defined patterns, and they must be given some degree of intelligence depending on the task they must complete. They can be used to perform tasks in a remote system to which they have been sent, and to bring the results later, when the task is finished, so that it is not necessary to keep the communication open during the whole process [17]. The reasons for this way of working are not only economical, but also practical: we can pay attention to other tasks while our agent is carrying out its (our) tasks. One example that shows the usefulness of this kind of system is the control of robots when exploring the surface of a planet. If the robot has to carry out some task in real-time and the delay in long distance communications, many times due to the narrow bandwidth that is available [12], [3], is unacceptable [18], [19], in this case, the use of mobile agents could solve the problem [20].

[1] Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s/n, 28660 Boadilla del Monte (Madrid), España.

Mobile agents involve several technologies, from communication systems to knowledge systems. Internet is the natural environment for mobile agents, but it is not the only one. These pieces of software collaborate in the network through a client-server model with distributed computation. We can design 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) or Distributed Common Object Model (DCOM) in order to have communication with different and separated processes and objects. All of these different programming techniques aim to provide mobility, flexibility and concurrence. Some authors claim that the only valid technique to develop mobile agents is RMI, since, as it is based on Java, it is the only real multi-platform option, given that different CORBA implementations are not 100% compatible.

In our research, we propose to 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 until we decide to take control of the situation. Given that many of the existing VEs are developed using different programming languages, partly due to performance reasons, for the moment the use of CORBA or RPC cannot be discarded.

3 MOBILE AGENTS INSIDE VIRTUAL ENVIRONMENTS

We propose the concept of a Mobile Intelligent Virtual Agent (mIVA) in order to be our representative in VEs. These environments could be Travel Agencies, Market Places, Virtual Stockbrokers, Virtual Libraries and so on.

First, let's see how mobile agents can be useful inside a VE. We must take into consideration that there are very different kinds of VEs: educational, medical, social, etc. Thus, mobile agents are not likely to have the same utility in all of them, and it may not even be a good idea to use mobile 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, both for the agent and for the VE. The same applies to the Intelligent Agents that populate the VE, since they must have an interface with which mobile agents can interact in order to give instructions and receive results.

3.1 Possible uses of mobile agents in Virtual Environments

Asking how a mobile agent is useful in a VE is like asking how 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.

In many occasions, an agent inside a VE is just an autonomous 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 will be able to accept some high level instructions from the user, and then decide which is the best way to carry them out [3].

So now the question to be answered is which of the actions that a mobile agent can perform 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.

Let's see some examples where the integration between mobile agents and VEs might proof useful.

3.1.1. Social VEs

One of the main uses of VEs nowadays is as social environments, where people just connect for the sake of chatting and meeting other people. Thus, the VE would act just as a three dimensional interface for a chat, with enhanced capabilities, but without any specific or useful purpose other than entertainment.

Then, what are the interesting aspects of this kind of VE for a user? As far as we have observed, mainly two: the users connected to it and the ambient of the VE, so these are the two aspects of which mobile agents will have to take advantage.

First, it may be quite interesting for a user to locate the representative of another person in order to 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 requested 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 find out their purpose and evaluate if any of them may have some interest for us. What's more, once we have confirmed that one of those VEs is interesting, we could send our agent back in order to perform some specific action in it instead of having to do it by ourselves. For example, out agent might queue to get some tickets to attend to an event where the number of seats (connections) is limited, so that we can connect later to it and enjoy the event, or we could send our agent to a virtual auction with instructions to bid for some of the objects on sale.

3.1.2. Games

Another possible use of mobile agents in VEs could be in Quake-like multi-user games, where users could customize and train their warriors in order to send them to a common server or to other player's machine where the agents would fight against each other.

Besides, agents could observe other agents fighting and learn their tactics and acquire more fighting skills by taking part in combats against them. In this case, the users could customize the learning algorithms used by their agent or even its initial tactical and fighting skills.

3.1.3. Advertisements

Exploring beyond the social uses of VEs, but still making use of the described functionalities, one of the purposes that will turn out to be very useful and used is advertising.

Advertisers make use of any media at hand to show their products and their enormous benefits for consumers. As such a potential medium, advertisers might send their mobile agents to look for interesting VEs where they could advertise their products. The reception of the VE could provide mobile agents with information about the purpose of the VE or the profiles of its visitors, so that they could decide if the VE is a good place to advertise their products. This feature might be much more interesting if users and agents had to pay to enter VEs, since they would not pay to enter a VE where the potential clients are not likely to be found.

3.1.4. Getting in touch

Let's imagine we are taking part in a virtual auction. A virtual auction doesn't provide many advantages over real auctions or even the auctions that currently take place in some web sites, but it does provide some. First, because people all over the world can take part in them in real time, and second, because it can recreate that particular atmosphere that auctions have and some people love. In this respect, virtual auctions are not that different from social VEs, and people would connect to them just for the emotion of taking part in an auction and, of course, for their interest in the objects being sold.

In this scene, a mobile agent might be useful as an observer of other people's actions. Thus, if it sees someone buying several pieces of Chinese pottery, the mobile agent of a salesman can get in touch with him in order to offer him some other similar pieces of pottery that the owner of the mobile agent sales. The agent and the buyer might then arrange a meeting between both parties where they could finish the operation.

This scenario can be translated into any other environment where it might be interesting to find people with certain characteristics.

3.1.5. The virtual shop

If you have ever bought some furniture for your house, you may have had to spend several days going from one shop to another until you have found the pieces you like for your house. There are some manufacturers who have already started to offer their products in the net, and even a few of them provide their clients with 3D models of their products, so that they can figure out how a table is like and whether it is what they are looking for or not.

Extending this recently used feature, it is not very far in the future when these manufacturers may have a virtual shop where they exhibit and sell their products.

Another fact that is starting to be common in the world of housing is architects providing their clients with 3D models of the houses they have bought or they are planning to buy, so they don't need to visit the house to see it and compare it with other houses.

Taking advantage of this facts, there are two possible uses a mobile agent may have in this scenario.

On the one hand, a user could send a mobile agent to search in the virtual shops what he is looking for. If the mobile agent doesn't find anything suitable for its owner's needs, he can ask other users or agents visiting the shop if they have found something similar during their trip, so that mobile agents and users can cooperate to find whatever they may need. Besides, there may be some agents or even users that play the role of the shop assistants, so they can help the customers find a product either in that shop or in a different one.

On the other hand, it may even be possible to make the mobile agent travel with the 3D model of its owner's house, so once the agent is in the shop it can try to decorate the house in order to see how it will look like or if the pieces of furniture have the appropriate size. Alternatively, a mobile agent can travel from the shop to the user's machine in order to perform the same activity.

One extension to this mechanism would be the organization of a tradeshow. In this case, the organizer would provide with the place to build up the tradeshow, and the participants would upload their stands and send their mobile agents to deal with the assistants. The users could attend to this event or send their agents to it in order to search for interesting goods, and the mobile agents sent by the vendors could travel to the clients' devices in order to give detailed information about their products.

3.1.6. Virtual services

Lets imagine now that we are organizing a workshop. In a close future, this kind of event will start taking place as a virtual meeting, so there will not be a physical place where the workshop takes place. Instead, either videoconference or, even better, VEs will be used to gather all the participants together, so it won't be necessary to travel anywhere to attend to the meeting.

If there is some colloquium where several participants may speak, it may be interesting to elaborate a transcription of the discussion in order to attach it to the proceedings of the workshop, and this task might be carried out by a piece of software.

However, since this is not something we would use quite often, we might prefer to rent this software instead of buying it. Thus, there could be a company that provided the services of a mobile agent that played the role of a secretary that would transcript the colloquium and would take some physical appearance so that the participants in the colloquium could interact with it in some way.

This solution would be more inexpensive than buying the software, the owner of the program would not have to worry about illegal uses of his application, since the mobile agent would not stay on the client machine after the service has finished, and the client would not have to worry about installation of the software, since mobile agents do not require installation.

This scenario is just an example of how mobile agents might travel from one VE to another one providing specialized services for hire.

3.2 The role of mobile devices

We propose an architecture where mobile devices can be used, firstly, to build mobile agents (mAs) and send them to be our representatives in different VEs (see Figure 1). When these mAs find the information required they would come back to our mobile device to inform us about the result of the assigned task.

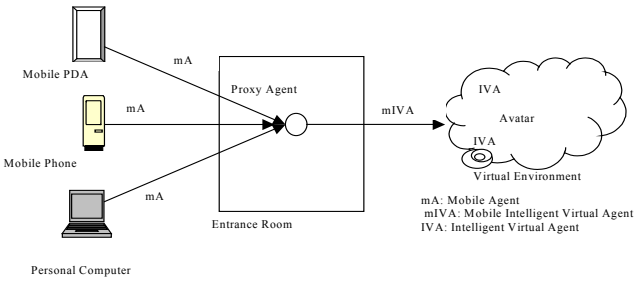

Figure 1. System Architecture

With this two-step process we can reduce the cost of transmission because we are using the media only when we are transmitting the mA to the VEs and when the mA comes back, but not while the mA works. Not only can we reduce the cost in terms of bandwidth, but also in economical terms.

There are a lot of mobile devices nowadays, more powerful every day, and we have to provide them with appropriate operating systems and useful interfaces. Useful in the sense of distribution ability, low cost in memory space , and, finally, low cost in communications, not only in economic terms due to the service and the telecom providers, but also in the amount of bandwidth that a sophisticated interface could require of the mobile device.

All new things are a challenge in some way. In the beginning, isolated computers appeared, and then, little by little, the need of interconnect these computers and with this, the development of

networks. These networks grew up to reach virtually all places but ever with a physical medium, like wires. Next, developers thought of wireless systems that could cover kilometre distance, but thinking more on saving investments in network infrastructures than on mobility . Mobile devices came to break this classic concept of a network, bringing up a more flexible use of the media.

These mobile devices work like an extension of a rigid network to a new type of devices.

When we are talking about mobile devices, we have to talk about distributed programming like a balance between the cost of process and the cost of used bandwidth. In this way we can think in mobile agents in the sense of intelligent distributed programming. One of the motivations behind the beginning of mobile agents was a more reasonable use of the communication media. Nowadays, mobile devices, like PDAs, are reasonably powerful in process and memory, but the increase in bandwidth has occurred really more slowly. It is technologically less hard to increase the process of a device than the performance of the network.

Application servers in the service provider should play the role of a receiver of mobile agents requesting a service, that could have been created and introduced into the network from devices physically linked to the network or from mobile systems..

3.3 Features of a Virtual Environment in order to accept mobile agents

Once we have seen how a mobile agent would 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.

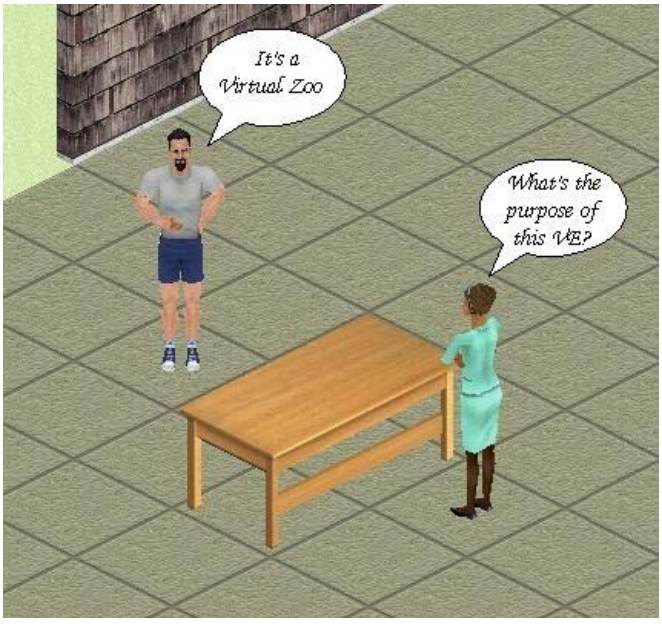

Figure 2. A Mobile Agent at the reception of a VE

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 about the purpose of the VE, users and agents connected to it, ongoing and future events to 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 (Figure 2 illustrates this idea). As we will explain 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 a VE, it won't interact directly with the VE, but by means of another agent, an Intelligent Virtual Agent (IVA). This will also be an advantage in terms of security, since the mobile agent will only be able to carry out the actions that the IVA allows it to do.

3.4 Features of an Intelligent Virtual Agent in order to be controlled by a mobile agent

As we have mentioned before, a standard interface that allows the communication between mobile agents and the IVAs that inhabit the VE has to be defined, which, as we can see in Figure 3, 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 very different purposes. For this to be possible, the communication must be general enough in order to grant 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, IVAs that are able to move in very different VEs taking advantage of all their features 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 virtual agent that visits the VE and interacts with the mobile agent 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.

- The possibility to receive VE-specific commands. For example, it 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, which would allow the agent to perform the VE-specific tasks. The use of ontologies would allow the agent to know the effect of these actions, and to decide whether it would be interesting to perform them or not.

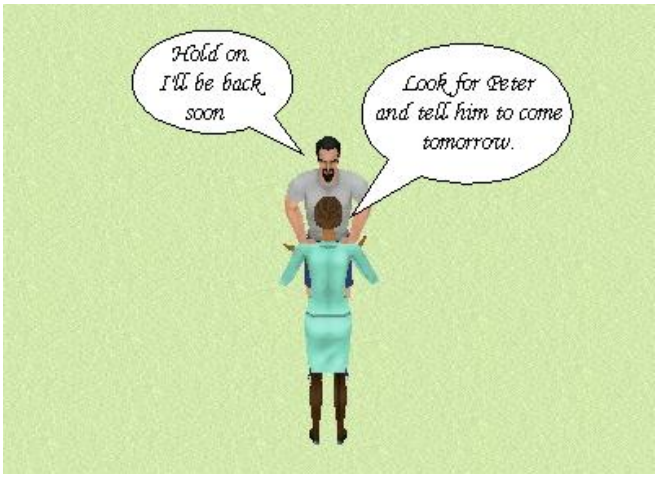

Figure 3. A Mobile Agent instructing its proxy Agent

3.5 How the whole system can work

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 will 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 will send the agent to a remote host with detailed instructions [21]: 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 (a mobile device) 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 action might be followed [21]: 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. 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 intelligent virtual agent (IVA) will be created in the VE, together with an avatar with a certain physical appearance, and it will be assigned to the mobile agent in order to 'guide' and represent the mobile agent inside the VE. All the communication between the mobile agent and the VE and its inhabitants will take place using the local agent as a proxy as if it were a 'normal' user. Figure 4 shows the difference between a normal avatar, directly controlled by a user of the VE, an IVA attached to and controlling an avatar, and the proposed situation, in which the mobile agent represents the user and is associated to an IVA, which in turn controls an avatar. The pair 'mobile agent+IVA' is what we call mIVA.

Since the mobile agent is associated with one of the IVAs that have been made to populate that VE, the mobile agent can give it the necessary instructions 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 IVA to look for it. Once it has been found, the mobile agent will ask, through the associated IVA, whatever the user has told it to find out, or it will let its owner 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 most 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 book our place in a virtual event where the number of attendants is limited or is so big that we want to get a good seat in order to see the show in detail.

But there will be possibly some actions that are so domain-specific that we will not be able to train our agent to perform them, since it would take so long 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. Since that features would not be very common in other VEs, there is no point in including them as standard features for all the VEs that accept the presence of our mobile agents.

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.

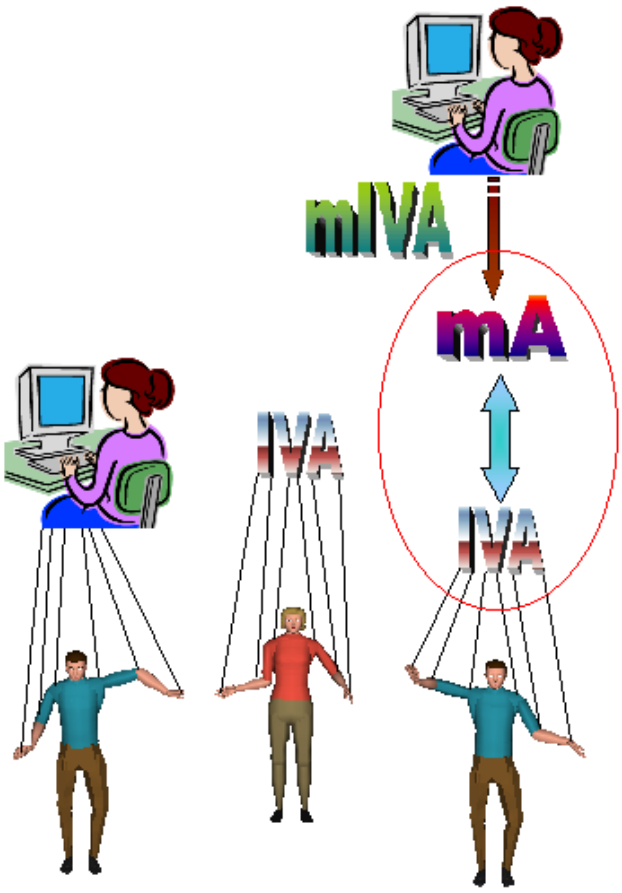

Figure 4. Avatars controlled by a mIVA, an IVA and a user

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.

4 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 [22], [23].

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 [24].

The basic idea is to integrate an agency with the VE, so when an agent arrives to the VE, it can immediately adopt a physical appearance and can be assigned a local intelligent virtual 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 about its existence.

Every time an agent arrives, it will supply the VE with information about the VEs it has already visited and will ask either for some information or for permission to enter the VE. The mobile agent will be informed of the ontologies to be used and necessary modules to be loaded in order to allow for a detailed communication with the proxy agent, so that specific actions can be carried out. These requests have been thought to be implemented using KQML [24], [25], [26], [27], which provides a standard language to construct them, although KIF [28] and Ontolingua [29] are also being evaluated..

When the mobile agent leaves the VE, the local IVA 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.

5 CONCLUSIONS

As we have mentioned before, although the system has not been implemented yet, we are currently working on it, and we believe that it is not very far in the future the moment when these technologies will be fully integrated and working in different kinds of VEs and also accessible via mobile devices.Of course, there is still a lot of work to be done, and some time 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 , Virtual Environments and Mobile Devices 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.

ACKNOWLEDGEMENTS

We would like to thank Ricardo Imbert for his support in the elaboration of this paper, without which this contribution would not have been sent to the workshop.

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-Verlag. 1996.

[2] G. Méndez, P. Pérez, A. De Antonio. "*An Overview of the Use of Mobile Agents in Virtual Environments*". Proceedings of the Third International Workshop on Intelligent Virtual Agents (IVA01),

LNAI 2190, Springer-Verlag. Madrid, Spain, September 10-11, 2001.

[3] 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.

[4] J. E. White, "Telescript technology: the foundation for the electronic marketplace", White Paper, General Magic, Inc. 1994.

[5] Clemments, T. Papaioannou, J. Edwards. ***"Aglets: Enabling the Virtual Enterprise".*** ME-SELA '97, p425.

[6] 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.

[7] M. Straer, J. Baumann, F. Hohl. "*Mole - A Java based mobile agent system*". ECOOP '96 Workshop on Mobile Object Systems, 1996.

[8] http://mole.informatik.uni-stuttgart.de/

[9] 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.

[10] 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.

[11] http://www.merl.com/HSL/Projects/Concordia/

[12] Maes, Guttman & Moukas. ***"Agent-mediated Electronic Commerce: A Survey".*** Knowledge Engineering Review, June 1998.

[13] R. Gray, "*Agent Tcl: A flexible and secure mobile agentsystem*", PhD thesis, Dept. of Comp Science, Dartmouth College, June 1997.

[14] 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.

[15] Nick Jennings et al. ***"Autonomous Agents for Business Process Management".*** International Journal of Applied Artificial Intelligence, 2000.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] T. Finin, Y. Labrou, and J. Mayfield. "*KQML as an Agent Communication Language*" Software Agents, MIT Press, 1997.

[27] 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.

[28] Knowledge Interchange Format (KIF). http://logic.stanford.edu/kif/

[29] T. Gruber. "*Ontolingua: A Mechanism to Support Portable Ontologies*". Stanford University, Knowledge Systems Laboratory, Technical Report KSL-91-66, March 1992.

[30] B. Jung, J. T. Milde. ***"An Open Virtual Environment for Autonomous Agents Using VRML and Java"***. VRML 99. Paderborn, Germany. 1999.

[31] T. Papaioannou, J. Edwards. ***"Manufacturing Systems Integration and Agility: Can Mobile Agents Help?"*** Special Issue of Integrated Computer-Aided Engineering, IOPress, January 2001.

[32] T. Papaioannou, J. Edwards. ***"Towards Understanding and Evaluating Mobile Code Systems".*** Proceedings of Cooperating Information Agents (CIA-2000).