Mobile Devices to Assist Cultural Visits

M. E. Bonfigli (*), *G. Cabri, L. Leonardi and F. Zambonelli* (#)

(*)Vis.I.T. Lab, CINECA Supercomputing Centre, Casalecchio (BO), Italy
E-mail: e.bonfigli@cineca.it
(#)Dipartimento di Scienze dell'Ingegneria, Università di Modena e Reggio Emilia, Modena, Italy
E-mail: {giacomo.cabri, letizia.leopardi, franco.zambonelli}@unimo.it

ABSTRACT
Traditional visits to environments interesting from a cultural point of view are based on technologies that do not permit flexibility and modification to the visit itself on the basis of the visitors' profile. Software agents running on personal mobile devices can overcome such limitations and lead to achieve personalised visits. The innovative technology based on mobile agents seems to well fit this scenario, where the agents on the mobile devices can interact with fixed HW/SW infrastructures. This paper proposes the reuse of several concepts from the mobile agent technology in the design of distributed applications based on mobile devices. In particular, coordination issues are addressed, and reactive tuple spaces are proposed as coordination means: thanks to their openness and flexibility, they well suit a scenario composed by mobile entities.

KEYWORDS: Mobile agents, Mobile devices, Assisted visit

INTRODUCTION

Cultural spaces are all those environments, outdoor or indoor, where some elements of cultural interest (*cultural goods* in the following) are present. A typical example of cultural space is a museum, but also a city with its monuments can be considered a cultural space. Currently, the visits to cultural spaces are based on technologies that, even if consolidated, do not allow the required flexibility to personalise a visit on the basis of different criteria, such as the personal tastes and knowledge, the already done paths, the time availability, and so on.

This work focuses on the possibilities of exploiting software agents running on mobile devices for the visit to cultural spaces. *Mobile devices* mean all those hardware devices that are autonomous and portable. They have a certain computational power that ranges from the lowest one of GSM mobile phones, which have a high degree of portability, to the highest one of the notebooks, which are less comfortable to be transported; in the middle we can find devices that have a medium computational power and a medium portability, such as the palmtops or the subnotebooks. Anyway, all of them have a graphical interface for the interaction with the user and an interface to exchange data with fixed infrastructures, for instance via an infrared port.

The *heterogeneity* due to the presence of several kinds of mobile devices calls for studying solutions that are so flexible as to adapt to the different situations. In particular, the same information must be shown by devices that can have different capability in terms of computation, portability and, above all, presentation.

This paper proposes the use of *agents* executing on mobile devices. Being on devices that dynamically change their environment, the agents can be considered *mobile* in their turn; so it is possible to reuse several concepts and results of the mobile agent technology. In particular, the paper will consider the coordination issue, since the interaction and the exchanging of information are fundamental activities in distributed applications based on mobile entities.

Starting from the existing HW and SW systems, we evaluate how to model the distributed application for the visit with mobile agents. In particular, we explore how much of the concepts and results of mobile agent technology can be reused in applications based on mobile devices. We claim that the coordination models based on tuple spaces grant the required adaptability to suit a general mobile scenario, no matter if the mobile entities are hardware or software. Further flexibility can be achieved by permitting the tuple spaces to react to the accesses in order to adapt their behaviour to the needs of applications or sites.

From the programming language point of view, the Java programming language is proposing as a fundamental tool for the development of applications in distributed and high heterogeneous environments, thanks in particular to the code portability on different platforms. Since the current implementations of the Java language require a significant amount of resources, its use in mobile devices must be carefully evaluated. A possible alternative that is gaining ground is the operating system WindowsCE, available for several palmtops. Obviously, it is at a different level with regard to Java, and it grants a more limited portability, but this operating system exhibits some interesting features that worth to be analysed.

The paper is organized as follows. Section 2 introduces the mobile agent technology and briefly discusses the coordination issue. Section 3 proposes the use of agents in mobile devices to assist the visit to cultural places. Section 4 concludes the paper and sketches future work.

MOBILE AGENTS

An innovative technology that is gaining interest in the distributed application area is the one based on *mobile agents* [5]. The mobile agent paradigm implies the definition of applications composed by proactive and network-aware entities (*agents*) capable of actively changing their *execution environment* by transferring themselves while executing (*mobility*). The current interest in distributed applications of this paradigm is widely justified by the advantages it provides over traditional approaches: *(i)* mobile agents can significantly *save bandwidth*, by moving locally to the resources they need, instead of requiring the transfer of possibly large amounts of data; *(ii)* mobile agents can carry with them the *code to manage remote resources* and they do not need the availability of specific servers on the remote node, leading to a more flexible application scenario; *(iii)* the execution of a mobile agent application does not require *continuous network connections*, because interacting agents can move to the same site when connections are available and interact without requiring further network connections. Consequently, *(iv)* the mobile agent paradigm intrinsically facilitates the integration of *mobile computing* systems in distributed applications: agents associated to a mobile computing device assume the role of mobile agents with regard to the fixed infrastructures.

In the distributed applications based on mobile agents, the flexibility is above all required with regard to *coordination* [7]. In fact, a fundamental activity in mobile agent applications is represented by coordination among agents and between agents and local execution environments. The distributed applications based on mobile agents are intrinsically collaborative and the definition of appropriate high-level coordination models to enable and facilitate interactions in the presence of mobility can have a strong impact on both application design and execution efficiency. The coordination models for the mobile agents that recently have been proposed with a high degree of success are those based on *tuple spaces* [1]. In these models, agent interactions occur via shared tuple spaces, residing on each hosting environment, which are used as information repositories where to locally store and retrieve messages. The most significant advantage of these coordination models is to achieve both spatial and temporal uncoupling in agent-to-agent interactions. When two agents need to exchange data, the sender agent can simply leave a message in the tuple space local to a given site, without needing to know neither where the corresponding receiver agent currently is, nor when it will read the message. When the receiver agent arrives at the given site, it can read the message left by the sender agent. This clearly suits a scenario where the position and the scheduling of the agents cannot be easily granted. In addition, being any agent-to-agent interaction forced to be performed via the tuple space, the hosting environment can easily monitor and control all interactions, thus leading to a more secure execution model. Furthermore, the accesses to the tuple space can be based on associative mechanisms, in a Linda-like fashion [2]: data can be retrieved from a tuple space on the basis of a pattern-matching scheme, without the need of explicitly naming the stored data but, instead, by accessing to it by its content. The well-known advantages of associative coordination models [4] particularly suit applications based on mobile entities. In fact, because of the dynamism and the openness of the environment, having a complete and updated knowledge of hosting execution environments may be difficult or even impossible, especially if it lacks a centralised control.

Further flexibility can be achieved by adding programmable reactivity to the tuple spaces: the behaviour of the tuple space can be dynamically programmed (by both administrators and agents) to be adapted to the peculiar needs of the sites or of the applications. The reactive tuple space models configure a technology suitable to facilitate the design and the development of distributed applications based on mobile agents. Moreover, they grant such flexibility as to result adequate to a scenario composed of mobile and heterogeneous entities.

ASSISTED VISITS TO CULTURAL SPACES

The adoption of a trendy technology, such as the mobile agent one, would permit to overcome the limitations imposed by traditional technologies. In particular, we propose the joining of the *physical* visit with a *virtual* visit, supported by innovative technologies. On the one hand, the *physical* visit permits the direct sight of a cultural good, which could be hardly substituted by the vision of the corresponding image, no matter if it is on paper or on video. On the other hand, the *virtual* visit permits to have more and updated information, which can be exploited adapting them to personal needs. The user that wants to visit a cultural space can therefore have a mobile device that guides him/her during the visit in a personalized, flexible and smart way.

The mobile device, put near the interesting cultural good, can exchange information with a fixed infrastructure. The software on the device (embedded or downloaded on demand), can explain the cultural good, selecting the pieces of information to be provided on the basis of the user personal profile; the profile could be defined by the user himself/herself or dynamically determined from previous visits. Moreover, keeping track of the done path, the software could, for instance, reference cultural goods already seen, and make some proposal about how to go on in the visit. The presence of more than one visitor must be taken into account, each of them with his/her own mobile device. In this case, they can "compete" for the visit to the same cultural good, or they can "collaborate" and decide to do the visit together, and in this case the software components loaded on the devices should agree on a path that can mediate the different needs and the different profiles.

We suppose that every cultural place is provided with a fixed infrastructure with HW/SW components to facilitate the local and remote visits. First of all, from the SW point of view, this infrastructure hosts a *database* that stores several kinds of information, above all about the cultural space, but it can also be considered as a repository for other useful information, such as the restaurant and hotels near to the cultural space. Then, there is a *Web server*, which is in charge of making the information in the database available for remote users, which, for example, want to know about the cultural space before performing the actual visit. From the HW point of view, the fixed infrastructure can provide a hardware component that permits the communication with the mobile devices near every cultural good (pictures, monuments, buildings, and so on); it can be considered as the local access point to the whole information system. From our point of view, the most interesting feature of the infrastructure is the capability of communicating with the mobile devices and exchange information, for instance via an infrared port.

There are two ways for exploiting the mobile agent technology in the applications based on mobile devices: the former one is to model the mobile devices as they were mobile agents; the latter one is to let mobile agents move from the mobile devices to fixed infrastructures and vice versa. The two alternatives are explained in the following subsections.

Agents Modeling Mobile Devices

The mobile agents can be used to model the presence of mobile devices in distributed applications [6]. In this case we can suppose that there is a fixed agent running on each mobile device. Since the device "moves", we consider that the fixed agent on a mobile device changes a part of its execution environment, so they have to face the peculiar issues of a mobile agent that changes its execution environment by moving from site to site. In this way, several concepts and results about mobile agent and mobile code [3] can be reused and exploited in taking into account mobile devices in distributed applications.

In the context of the assisted visit, a significant issue that must be taken into account is the exchanging of information between the mobile devices and a fixed infrastructure. Moreover, in the case that several users are attending the visit with their own mobile device, that fact that they can compete or collaborate implies that the exchanging of information can occur also among the mobile devices. Often, more than the simple exchanging of information is required, for example if synchronisation is needed, so we can talk about *coordination* of the involved entities, let them be fixed or mobile. In all cases, we propose the use of coordination models based tuple spaces. The fixed SW infrastructure provided with each cultural space can be considered as a site where the information useful to agents is stored in the form of tuples. Thanks to the associative mechanism, agents in mobile devices can retrieve such information in a flexible way, as explained in Section 0. In this case, the reactivity of tuple spaces can be used to recognise the user/device that has performed the request, and to prepare the information tuple to be returned on the basis of the user profile and the device capabilities. This leads to simplify the code of the agents and facilitate the programming task.

Agents moving from/to Mobile Devices

The latter way to exploit the mobile agent technology is to let personal agents move between the mobile devices and the fixed infrastructure provided with each cultural space. Mobile agents can perform their task (selecting information, for instance) also when there is no communication between the mobile device and the fixed infrastructure. For example, we can consider the users' home site, from which users can see a preview of the actual visit by means of a browser, and personalise it, for example by choosing the more appropriate path. In this way, users can also exploit several services, which can be directly related to the visit, for example the users can book visits, or even general services, such as the booking in a restaurant that is near the cultural space and is cheap enough for his/her funds. These services can be exploited *on-line* by a browser, but they can also exploited *off-line* by sending to the cultural space site an agent with the appropriate information, such as the preferred day for the visit, the range of prices for the restaurant, and so on. The same agent can negotiate the visit with the cultural space manager, and also with other agents to avoid too many people in the same place at the same time, allowing a more comfortable visit to all participants.

CONCLUSIONS AND FUTURE WORK

This work proposes the use of the mobile agent technology to assist the visits to cultural spaces. Users provided with mobile devices can be assisted by agents that not only explain the seen cultural goods, but also organise the entire visit on the basis of the users' profiles and provide the access to several related services. Sophisticated agents can perform more complex tasks on behalf of the users that sent them.

In context of virtual visits, the use of virtual reality technologies can improve the usability of the applications [8]. In fact, the users can access cultural goods that risk the damage, or which no longer exist, or of which we have only some fragments. Moreover, whole virtual environments can be built by putting together "objects" that are actually in different places and by simulating possible transformations in the time.

A future research direction can be a more strict integration with the Internet world, and in particular with the Web, so as to let users exploit different and complex services, also not directly related to the cultural space from remote.

Putting together the virtual reality and the Internet world, it is interesting to investigate the definition of a virtual-reality language for the description of cultural goods. Such language should be powerful to permit the description of complex picture or monuments, but at the same time it must be simple enough to fit both simpler mobile devices and slow Internet connections.

ACKNOWLEDGEMENTS
Work supported by Italian Ministry for University and Science and Technology Research (MURST) within the project framework of "MUSIQUE: Infrastructure for QoS in Web Multimedia Services with Heterogeneous Access" and by the University of Modena and Reggio Emilia with a fund for young researchers.

REFERENCES

1. G. Cabri, L. Leonardi, F. Zambonelli, "Mobile-Agent Coordination Models for Internet Applications", IEEE Computer Magazine, Vol. 33, No. 2, pp. 82-89, February 2000.

2. N. Carriero, D. Gelernter, "Linda in Context", Communication of the ACM, Vol. 32, No. 4, pp. 444-458, 1989.

3. A. Fuggetta, G. Picco, G. Vigna, "Understanding Code Mobility", IEEE Transactions on Software Engineering, Vol. 24, No. 5, pp. 352-361, May 1998.

4. D. Gelernter, N. Carriero, "Coordination Languages and Their Significance", Communications of the ACM, Vol. 35, No. 2, pp. 96-107, February 1992.

5. N. M. Karnik, A. R. Tripathi, "Design Issues in Mobile-Agent Programming Systems", IEEE Concurrency, Vol. 6, No. 3, pp. 52-61, July-September 1998.

6. A. L. Murphy, G. Roman, G. P. Picco, "Software Engineering for Mobility: A Roadmap", International Conference on Software Engineering, Limerik (IR), June 2000.

7. A. Omicini, F. Zambonelli, "Coordination for Internet Application Development", Journal of Autonomous Agents and Multi-Agent Systems, Vol. 2, No. 3, pp. 251-269, Sept. 1999.

8. K. H. Veltman, "Frontiers in Electronic Media", Interactions – New Visions of Human-Computer Interaction IV, 4, pp. 32-64, July 1997.

ABOUT THE AUTHORS

Maria Elena Bonfigli graduated with honors in Computer Science from the University of Bologna and received her Ph.D. in "History and Computing" from the University of Bologna. Since 1998 she has been collaborating with CINECA (InterUniversity Supercomputing Center), in a first time as external visitor and now as a research fellow. Her research interests include Virtual Reality and Human Computer Interaction applied to Cultural Heritage.
E-mail: *e.bonfigli@cineca.it*

Giacomo Cabri is a contract research associate in computer science at the University of Modena and Reggio Emilia. He received a Laurea degree in electronic engineering from the University of Bologna and a PhD in computer science from the University of Modena and Reggio Emilia. His research interests include tools and environments for parallel and distributed programming, wide-scale network applications, and object-oriented programming.
E-mail: *giacomo.cabri@unimo.it*

Letizia Leonardi is an associate professor at the University of Modena and Reggio Emilia, where she teaches basic and advanced computer science courses. She received a Laurea degree in electronic engineering and a PhD in computer science from the University of Bologna. Her research interests include object-oriented programming environments; parallelism and distribution issues, especially as they apply to object systems; and design and implementation of coordination infrastructures for agent systems.
E-mail: *letizia.leonardi@unimo.it*

Franco Zambonelli is an associate professor in computer science at the University of Modena and Reggio Emilia. He received a Laurea degree in electronic engineering and a PhD in computer science from the University of Bologna. He is a member of the IEEE and ACM. His research interests include parallel, distributed, and Internet programming; distributed algorithms for fault-tolerance; and debugging.
E-mail: *franco.zambonelli@unimo.it*