Tailored Distributed Accesses to Cultural Heritage

Maria Elena Bonfigli
Vis.I.T. Lab
CINECA Supercomputing Centre
Via Magnanelli 6/3 – Casalecchio di Reno (BO) – ITALY
Phone: +39-051-6171573 – Fax: +39-051-6137273
e.bonfigli@cineca.it

Giacomo Cabri
Dipartimento di Scienze dell'Ingegneria
Università di Modena e Reggio Emilia
Via Vignolese, 905 – 41100 Modena – ITALY
Phone: +39-059-2056143 – Fax: +39-059-2056126
giacomo.cabri@unimo.it

Abstract

Cultural heritage and information technologies are a joint venture that is gaining ground more and more, and will soon be spread in the Web environment. In this context, this paper proposes a distributed architecture to access cultural information. The main features of such architecture are a deep integration with the Web environment and a high degree of flexibility and autonomy, which permits the customization of the accesses on the basis of different criteria, in a transparent and automated way. Our architecture is based on innovative technologies such as mobile agents and active proxy servers. They permit to build up virtual visits through cultural heritage, which can be tailored on the basis of users' profiles and devices, and can be enhanced by the availability of distributed services such as the e-commerce.

Keywords*: Mobile agent, Proxy server, Dublin Core, Cultural heritage, Helped visit*

1 Introduction

The impact of culture on the development of new industries is one of the most crucial question facing the Information Society: *cultural heritage* is the platform for the development of new industries and new employment opportunities and the cultural economy is fundamental to the creation and expansion of the global Information Society.

So the information technologies can be fruitfully exploited to permit a wide access to the cultural heritage information. On the one hand, people can perform virtual visits from whatever they are in the world. On the other hand, physical visits can be supported by innovative technologies that provide enhanced services and make the visit more comfortable.

If we consider the information about cultural heritage spread over the Internet such as in a grid, it could be very difficult for a user to deal with all the cell of the grid without a loss of time; often, (s)he is confused by the great amount of information (see Figure 1). So, we propose a well-defined "point of access", which can be very helpful to overcome the information overload and to lead the user in fruitful accesses to the information. This point of access is more than a search engine, because it is in charge not only of selecting the required pieces of information, but also of organizing them, and of adapting them to the user preferences.

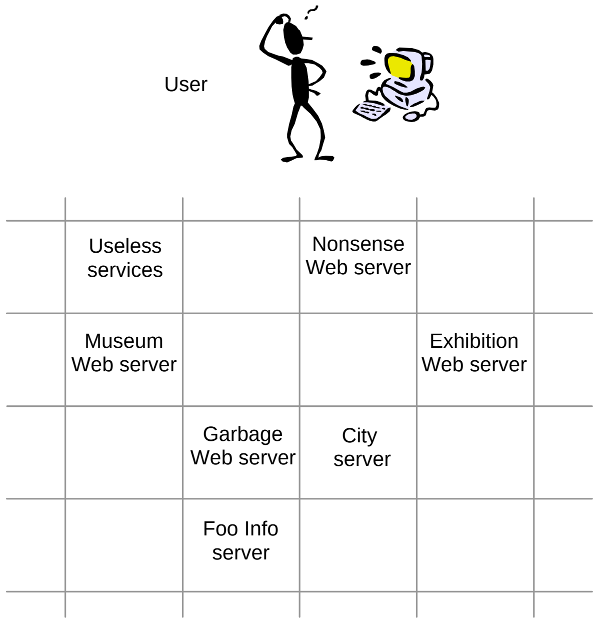

Figure 1. The user is overloaded by the amount of (often-useless) information

In this paper we focus on a distributed architecture that promotes "points of access" in order to permit virtual visits from remote. In particular, we spent an effort towards the exploitation of existing standards and distributed architectures.

Distributed approaches in the area of cultural heritage have already been proposed [Moe98] and focused on the capability of using a standard visualization protocol to display images and animation of cultural elements in Web pages. We propose a deeper integration with the Internet – and in particular with the Web – and more flexibility and autonomy that lead to the capability of automated customization of the accesses on the basis of different criteria. These goals can be achieved by means of a support that is well integrated in the Web and permits the reuse of existing applications. To this purpose, we exploit *PROOF* [CabLZ99, CabLZ00c], a proxy-based architecture that permits the development of distributed and interactive applications in the Web, starting from the existing servers and browsers. A further degree of flexibility and autonomy can be achieved by exploiting *mobile agents*, autonomous software entities that can act on behalf of users while roaming networks [JenW98, KarT98]. With regard to the grid of information previously sketched, a PROOF server can be exploited as the "point of access", in order to help users in accessing information fruitfully.

This paper is organized as follows. Section 2 presents PROOF, the distributed architecture that is exploited in our system. Section 0 shows the architecture of the distributed system that manages the access to the cultural heritage. Section 4 discusses the system's features. Section 5 concludes the paper and sketches the future research directions.

2 The PROOF Architecture

PROOF [CabLZ99, CabLZ00c] is an architecture designed to provide a mean to enrich the Web with computational capabilities without requiring significant modifications to current servers and clients. In fact, PROOF relies on the concept of *proxy server*, which stands in the middle between servers and clients. While traditional proxy servers are mainly used to provide cache functions, PROOF is much more flexible and can embody several different functionalities. Within the PROOF proxy server, any kind of computation can be enclosed, such as caching and dynamic production of HTML pages. Synchronous interactions between the clients and the proxy server can be enabled by letting PROOF insert specific applets, called *control applets*, into the pages that it provides to clients (browsers), also enabling communication among people connected to the Internet via the proxy server. Moreover, such a kind of proxy server can become a workplace open to the Internet, where the cooperation between clients can occur without forcing clients to exit the workplace when accessing generic Web servers.

PROOF is based on a modular architecture, composed by a *framework* and several application *modules* (see Figure 2). The framework provides the basic proxy functionalities, such as the connections with the client browsers and with the Web servers, the user identification and authentication; moreover it permits the embodying of different application modules. Each module implements the behavior of one specific application or functionality.

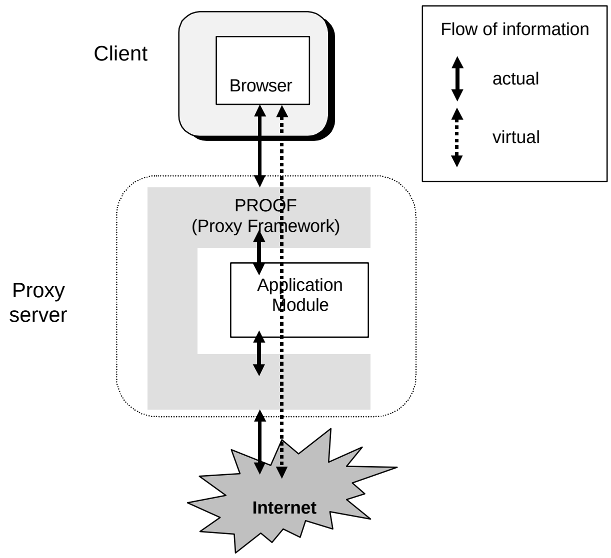

Figure 2. The application-dependent module defines the proxy server behavior

PROOF allows loading more than one module at a time, so as to permit one single proxy to be exploited by several clients (browsers or agents) for different applications. A demultiplexing software level permits to distinguish the different connections. PROOF makes also available a *cache* feature, which can (or cannot) be enabled by the single loaded module, on the basis of the specific application requirements.

The PROOF architecture is very general and it is not tightly bound to any specific application because it is based on the implementation of a framework that offers general-purpose application-independent functionalities. Different application modules can be implemented with a limited coding effort and easily installed within the proxy-framework.

2.1 Mobile-Agent Capabilities

The PROOF architecture integrates the capability of hosting *mobile agents*. They can significantly improve the design and the development of Internet applications thanks to their characteristics. The *agency* feature [JenW98] permits them to exhibit a high degree of autonomy with regard to the users. The *mobility* feature [KarT98] takes several advantages in a wide and unreliable environment such as the Internet. First, mobile agents can significantly save bandwidth, by moving locally to the resources they need and by carrying the code to manage them. Moreover, mobile agents can deal with non-continuous network connection and, as a consequence, they intrinsically suit mobile computing systems.

Therefore, mobile agents, on behalf of users, can install specific modules to give the proxy server an application-dependent behavior. Since agents are autonomous, the user can give them a possibly high-level task to carry out, and they can proactively search for the needed module(s), perhaps by mean of negotiation with module providers; once they have found the module(s), they can search for the most appropriate – also in terms of costs – proxy server where to install the module(s). The user is then notified of the proxy server (s)he has to use in order to exploit the needed functionalities. As a further advantage, this extension permits to give other entities – besides people – the access to interactive Web applications. Users can think of not participating directly to an application that requires repetitive actions; instead, they can rely on one or more software agents that act in behalf of them. For example, if a user is interested in buying a good, (s)he can delegate to a software agent the tasks of comparing the different offers, and of negotiating the final price. In this context, the exploitation of mobile agents can make the PROOF architecture very open and flexible.

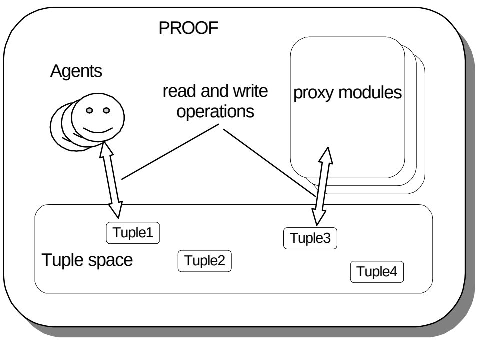

Figure 3. The tuple space integrated in PROOF

The presence of agents introduces several interaction and synchronization issues, and calls for the presence of appropriate coordination models to permit interactions, collaborations, negotiations and also competitions at a high level. Since *programmable tuple spaces* – based on the Linda model [AhuCG86] – are recognized to be a powerful coordination means that well suits Internet applications [Cia98], in particular those where mobile agents are involved [CabLZ00a], they are exploited in PROOF. Tuple spaces are used to store, retrieve and exchange information in a simple and standard way (see Figure 3); the programmable reactivity increases flexibility and permits to add whatever computation in the coordination media.

3 The System Architecture

We exploit PROOF to build up a distributed architecture where the PROOF proxy server is a point of access to different repository of information (see Figure 4), which are reached by mobile agents.

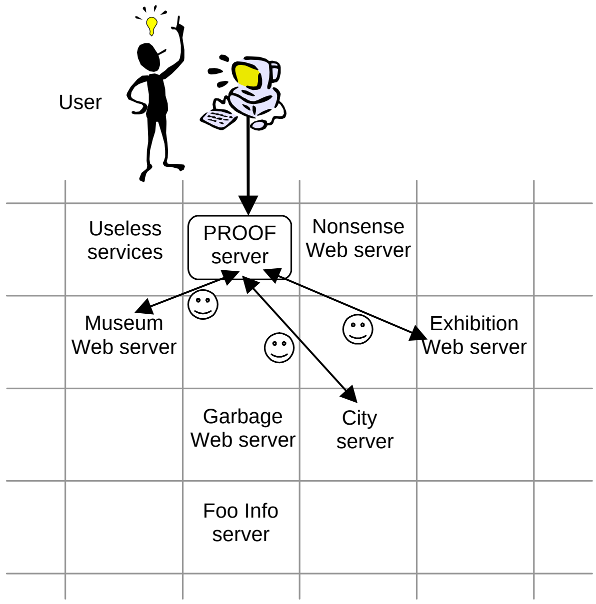

Figure 4. The user exploits a PROOF proxy server as a point of access

The two main aims of our architecture are *standard compliance* and *reuse of existing components*.

The first standard we adopt is the Web and, in this context, the PROOF architecture permits to reuse existing browsers and servers. In fact, in our architecture each cultural place – intended as a place that contains cultural heritage elements – is provided with a standard Web server that makes information available. Such information can be of different kinds, and in particular it is related to the description of the elements that can be visited. As explained later, the information is stored using a standard that permits to describe several historical and artistic issues, to allow detailed visits. Moreover, the server makes available also graphical information that is used to present images and animations to the user. The site of the cultural place may also have an information system that provides different services, such as the booking of the visits, possibly with a Web interface. From the user side, the only required component is a standard browser that supports Java; this permits to enable the control applet of the PROOF architecture.

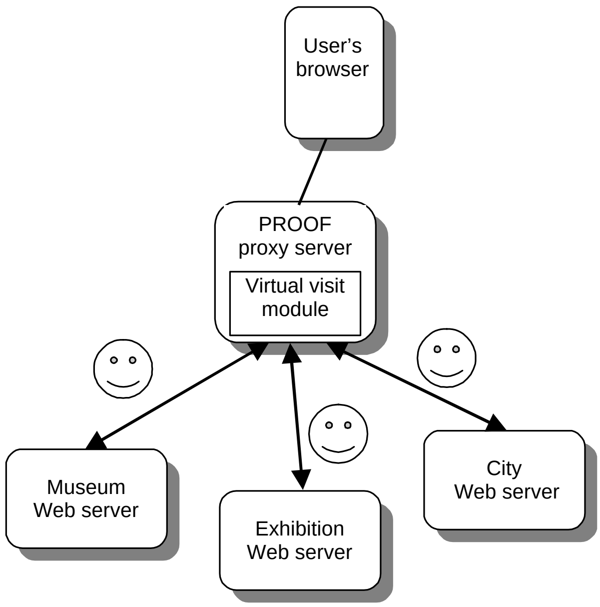

Figure 5. The distributed system architecture

The system works as shown in Figure 5: the user configures her/his browser to use a PROOF proxy server where a module for the virtual visit has been loaded. At the beginning of the session, a login page is provided to identify and authenticate the user. Then, the module inserts the control applet in all the sent Web pages and takes control over them. Mobile agents are exploited to delegate the operation of retrieving interesting information. Whenever a Web page is required, the module filters it to tailor the provided information, by either adding objects or simplifying the page. Moreover, the module can dynamically create pages that do not actually exist, to meet the user's requirements. For example, it can create a page that contains images of cultural places of the same area, with the links to the corresponding Web servers. As explained later, such "resuming" pages can be built on the bases of different criteria, not only the geographic one.

For simplicity's sake, in the following we refer to a single proxy server. However, the real architecture can be composed of several proxy servers, to distribute the load and to make special-purpose services available. For the load-balancing aim, a trivial solution is to provide several "points of access" in the form of several proxy servers, delegating the choice of the preferred one to the user. In a more sophisticated solution, a well-defined proxy can be in charge of accepting requests, finding the best proxy server, and then setting the user's browser to use such proxy server, by exploiting agents. In the other case we suppose that the proxy servers differentiate each other by the available services, and an agent is in charge of searching the proxy server that has all the services required by the user or has the best trade-off between costs and provided services. In both cases, a federation of proxies can be made up, to better answer the users' requests.

4 Features

We suppose that the Web sites describe art works and their digital reproductions using *Dublin Core* records. The Dublin Core Metadata standard (DC in the following) is a fifteen-element metadata set, originally conceived for author-generated description of Web resources [DubC]. Recently it has gained the attention of formal resource description communities such as museums and libraries due to the fact that: (i) it is useful to describe artifacts and associated information resources in the museum community and (ii) it is particularly simple to learn and easy to implement (using a basic XML syntax) [CIMI].

Cultural heritage information available through the Internet includes mainly multimedia data like written texts, images, video and audio files that describe art works in permanent collections and temporary exhibitions [Vel97, Vel97b]. Moreover the advent of virtual reality Web technologies (like Quick Time VR and Virtual Reality Modeling Language) enables the user not only to access multimedia data, but also to visualize and interact with 3D objects reproducing art works, galleries, museums, churches and other cultural monuments and to do virtual walkthroughs. This can be considered very interesting in order to visualize sculptures, buildings, and archaeological finds in which concepts like real volume and interactivity with the user are very important [BonG00].

Dublin Core enables to describe in a standard way all these different types of surrogate resources [BonCG00] but also both original resources like buildings, paintings, sculptures, and art works in general.

In our system, the use of this standard grants a high degree of interoperability and in particular enables the Virtual Visit Module to allow the user to obtain cultural heritage information tailored on the basis his/her own interests:

- enabling the creation of a collection of information coming from different sites;
- suggesting cultural paths to explore cultural heritage

contents distributed on the Web;
- providing the exploitation of related services.

4.1 Tailored Information collection

Interacting with our system, the user can look for art works related to specific keywords, historical periods, geographic locations or authors (all these data are specifically described by DC metadata elements). Moreover (s)he can decide to search for a specific type of digital reproduction, i.e. images rather then 3d models or videos according to her/his preferences and to available HW/SW components and bandwidth. Finally, the system enables the user to customize also the presentation of original or surrogate resources descriptions, visualizing only a selection of the fifteen DC metadata elements. The result of these interactions is a tailored collection of information that constitutes also the basis for further elaborations (see subsections 4.2 and 4.3)

In order to provide the described features, the Virtual Visit Module:

1. selects the DC records from different sites on the basis of the user's interests and choices;
2. builds customized XML files on the basis of the DC subset selected by the user;
3. visualizes them using appropriate XSL style-sheets.

The more flexible way to perform the customization task is to exploit mobile agents. Retrieving the Virtual Visit Module and choosing a PROOF server where to install it are the preliminary tasks that can be delegated to agents. This first step may involve also negotiation for buying/renting the module and for the costs of proxy resources. Then, the first time the user connects to the PROOF server, a mobile agent that knows her/his preferences can set up an appropriate profile in the PROOF server. In this case, the agent sets generic information about the user, such as the degree of experience (ranging from "simple tourist" to "archaeologist"), the language, and the available bandwidth. After every virtual visit, a mobile agent collects feedback information from the user – both explicitly via a GUI and implicitly by recording the choices made during the visit – and goes to the PROOF server to update the profile. Moreover, such update action records also the visited cultural spaces and art works, which will be taken into considerations in further visits, as shown in the next subsection. The use of mobile agents can avoid repetitive and boring tasks to the user, enhancing the degree of autonomy of the system. Users with a low bandwidth or with discontinuous connections to the Internet can also exploit the fact that mobile agents act on behalf of them while they are not connected to the network.

4.2 Tailored Paths

If we refer to real visits, we find out that usually they are not limited to a single museum, or a single cultural place, but they often provide a cultural path composed of some steps. For example, if a user performs a real visit to Rome in Italy, (s)he is likely to choose a path that includes *Via dei Fori Imperiali*, the *Foro Romano* and the *Colosseo*, because they are in the same area and are related to the ancient roman empire culture. Similarly, a virtual cultural path can be constructed on the basis not only of the geographical proximity, but also of other motivations, such as personal interests, historical affinity, and so on. A trivial task could be making up paths that contain related cultural elements. But our architecture goes further, and proposes two main facilities in the context of cultural paths.

The former facility is related to the capability of referring other already-seen cultural elements. For example, when a picture is shown in the browser, the user can be informed about other pictures of the same author seen in a previous (virtual) visit to another place. This is permitted by the fact that the module installed in PROOF keeps track of the visit of every user. Thanks to the representation of the information via the Dublin Core standard, connections among cultural elements can be build on the basis of the author, the period, the subject, and so on, achieving a graph that can be explored following the user preferences stored in the user profile.

The latter facility provides dynamic paths, where the next place to visit is decided after the latest one. If a visit to a museum of medieval pictures was boring for the user, it is better not to propose a visit to a similar museum, while if the visit was exciting, the system should search for a similar one to be proposed to the user. This facility is a bit more difficult than the previous one, because often the feedback of the user is not so sharp, and the system has to accomplish the user's desires. Also in this case, agents are used to obtain a feedback from the user and to go to the proxy server in order to tune the user profile.

4.3 E-Commerce

E-commerce will be integrated by installing a specific module in PROOF. Besides the trivial use of selling books, catalogues and gadgets, we intend to perform two main exploitations.

The former one relates to the services that are complementary to the visit itself. After virtual cultural tours, users can decide to perform a real visit in the physical space where art works are collected. In this case, the e-commercial module installed in PROOF should build up a real cultural path that can be exploited to retrieve all those services connected to the visit, such as restaurants, public transport, and so on. The module should generate a Web page containing information about the found services and, when available, a link to the Web server of the service, so that the user can get more information and even book online. In this context, the presence of agents is very useful, because the user can send an agent that is in charge of negotiate the different steps of the path, starting from a given budget and a range of available time. The agent evaluates different proposals and chooses the one that best fit its user's preferences and availability.

The latter case takes into consideration that the proxy modules consume resources of the server site where they are hosted. We can figure out that such services are charged to the user, on the basis of the quality of service the servers grant. The user can choose the server that best fit her/his preferences, while not exceeding her/his budget. Since the negotiation is quite boring, it can be delegate to a mobile agent, which knows the preferences and the budget of the user, and roams to several servers to negotiate the cost of the services, for example via auctions [CabLZ00b, SanH00]. Once the server is chosen, the agent configures it with the user profile and advises the user that the virtual visit can start. Moreover, during the virtual visit, the agent can monitor the cost of the other servers in the network, and can dynamically move the module to a cheaper server.

5 Conclusions and Future Work

This paper has shown how the distributed access to cultural heritage can be tailored and integrated in the Web. The use of an enhanced proxy-based architecture, such as PROOF, can provide several advantages in terms of:

- *customization* of the accesses to the information on the bases of different criteria, among which the personal preferences and interests of the users and the capability of the SW/HW components;
- *reuse* of existing SW components, so that users can access by using their browsers and do not need additional programs;
- *adaptability* to existing standards, such as the Dublin Core;
- *distribution* of computation over (wide) networks.

In this context, mobile agents play a fundamental role, since they can significantly improve the architecture in terms of:

- *delegation of tasks* such as the retrieval of pieces of information;
- *efficiency* in the distribution of load;
- *advanced interaction* among the different components.

With regards to future work, there are some research directions that are to be explored.

The first one is related to *mobile devices*. We envision that users will be provided with personal devices such as PDAs, palmtops, or even mobile phones, which help them in the visit to cultural heritage. It is interesting to model visit applications that take into account the help these devices can provide to user, also in connection with the virtual visits performed via Web.

The second direction concerns the *federation of proxies*, intended as a grid where each cell id a proxy server. It could be interesting to study different forms of interaction and collaboration among a grid of proxy servers, also taking into account the role played by agents, that could move from one server to another to easy the interaction task.

A further direction explores the concept of *digital cities* [DigC00], i.e. the digital information infrastructures that are becoming pervasive in today's cities. We can figure out a deep integration of our architecture in a more general infrastructure that provides not only information, but also several kinds of services next to the place where people live.

Finally, *social conventions* are to be imposed if we consider the distributed system as a place where people and delegated agents meet and interact: this leads to the fourth research direction, related to the general issue about how to impose social rules in a wide open environment such as the Internet [ZamJW01].

References

[AhuCG86] S. Ahuja, N. Carriero, D. Gelernter, "Linda and Friends", IEEE Computer Magazine, Vol. 19, No. 8, pp. 26-34, August 1986.

[BonCG00] M.E. Bonfigli, L. Calori, A. Guidazzoli, M.A. Mauri, M. Melotti; "Tailored virtual tours in Cultural Heritage worlds", ACM SIGGRAPH2000, New Orleans, July 2000.

[BonG00] M.E. Bonfigli, A. Guidazzoli, "A WWW Virtual Museum forimproving the knowledge of the history of a City", in Virtual Reality in Archaeology, (J.A. Barcelo, M.Forte, D.H. Sanders Eds.), ArcheoPress, May 2000.

[CabLZ99] G. Cabri, L. Leonardi, F. Zambonelli, "A Proxy-based Framework to Support Synchronous Cooperation on the Web", Software, Practice and Experience, Vol. 29, No. 14, pp. 1241-1263, 1999.

[CabLZ00a] 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.

[CabLZ00b] G. Cabri, L. Leonardi, F. Zambonelli, "Auction-based Agent Negotiation via Programmable Tuple Spaces", Proceedings of the 4th International Workshop on Cooperative Information Agents (CIA) 2000, Lecture Notes in Computer Science, No. 1860, Boston (USA), July 2000.

[CabLZ00c] G. Cabri, L. Leonardi, F. Zambonelli, "A Web Infrastructure for People and Agent Interaction and Collaboration", Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE CS Press, NIST (USA), June 2000.

[Cia98] P. Ciancarini, R. Tolksdorf, F. Vitali, D. Rossi, A. Knoche, "Coordinating Multi-Agents Applications on the WWW: a Reference Architecture", IEEE Transactions on Software Engineering, Vol. 24, No. 8, pp. 362-375, May 1998.

[CIMI] Consortium for the Computer Interchange of Museum Information (CIMI), Guide to Best Practice: Dublin Core (DC 1.0 = RFC 2413), final Version, 12 August 1999.

[DigC00] "Digital Cities", T. Ishida, K. Isbister, eds., Lecture Notes in Computer Science No. 1765, Springer-Verlag, 2000.

[DubC] Dublin Core Metadata Element Set, Dublin Core home page, http://purl.org/dc.

[GelC92] D. Gelernter, N. Carriero, "Coordination Languages and Their Significance", Communications of the ACM, Vol. 35, No. 2, pp. 96-107, February 1992.

[JenW98] N. R. Jennings, M. Wooldridge, eds., "Agent Technology: Foundations, Applications, and Markets", Springer-Verlag, March 1998.

[KarT98] 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.

[Moe98] W. E. Moen, "Accessing Distributed Cultural Heritage Information", Communication of the ACM, Vol. 41, No. 4, pp. 45-48, April 1998.

[SanH00] T. Sandholm and Q. Huai, "*Nomad*: Mobile Agent System for an Internet-Based Auction House", IEEE Internet Computing, Special issue on Agent Technology and the Internet, Vol. 4, No. 2, pp. 80-86, March-April 2000.

[Vel97] K.Veltman, "Frontiers in Electronic Media", Interactions Journal of the ACM, New York, pp. 32-64, July-August 1997.

[Vel97b] K.Veltman, "Hypermedia: New Approaches to Cultural Heritage and Knowledge", The Global Village Conference, Vienna, pp. 1-13, February 1997.

[ZamJW01] F. Zambonelli, N. R. Jennings, M. Wooldridge, "Organizational Rules as an Abstraction for the Analysis and Design of Multi-agent Systems", Journal of Software Engineering and Knowledge Engineering, to appear, 2001.