Integrating Mobile Agent Infrastructures with CORBA-based Distributed Multimedia Applications

Paolo Bellavista, Antonio Corradi
DEIS - Università di Bologna
{pbellavista, acorradi}@deis.unibo.it

Domenico Cotroneo, Stefano Russo
DIS - Università di Napoli Federico II
{cotroneo, Stefano.Russo}@unina.it

Abstract
The increased computing power and the enhanced connectivity of current open computing systems are encouraging the deployment of new classes of services both centered around dynamically changing user requirements and based on the exploitation of the Internet infrastructure. Distributed Multimedia Applications (DMAs) are a typical class of services with challenging requirements in terms of resource demand, dynamicity and QoS adaptation. The paper claims that distributed objects and mobile agents can complement each other to provide a flexible middleware for DMAs, and describes the case study of MADAMA (Mobile Agent-based Distributed Architecture for Multimedia Applications). MADAMA adopts mobile agents to simplify the distribution of service control and to provide location-aware adaptability. In addition, MADAMA is compliant with CORBA to achieve large accessibility and interoperability.

1 Introduction
The possibility to interconnect heterogeneous resources, mainly due to the widespread diffusion of the Internet, and their increasing computing capabilities have recently stimulated the design and implementation of new classes of services for a wide range of applications. Emerging services require the possibility to exchange large amounts of data, e.g., multimedia flows, and suggest the adoption of mechanisms to obtain a desired level of Quality of Service (QoS), e.g., reserving/adapting to the communication bandwidth that is currently available for multimedia flows. In any case, new services force to consider new requirements, e.g., coordinated forms of code and data distribution, flexibility, QoS monitoring and adaptation, hard to satisfy within a standard best-effort communication network such as the Internet.

The paper focuses on the services for Distributed Multimedia Applications (DMAs), e.g., computer-supported cooperative work, video conferencing and video-on-demand. This class of services has recently raised new interest because of the technology advances in several areas, from networking to media processing, that have made possible for end-users highly interactive and real-time access to remote resources. Traditional DMAs are tied to proprietary and closed implementations, generally accessible only via proprietary clients and with a low degree of extensibility and customizability. In addition, they cannot usually adapt to the run-time current conditions in involved end-systems and in intermediate hosts of the network path.

Several research activities agree on the proposal of middleware solutions to simplify the design and implementation of flexible network services that achieve high accessibility, customizability, extensibility and adaptation [1-3]. It is widely recognized that the traditional Client/Server (C/S) approach lacks the flexibility needed to satisfy the new network- and user-centric requirements. Some researchers claim the necessity of new flexible programming paradigms based on dynamic migration of both code and execution state, such as in the Mobile Agent technology (MA) [4,5]. In addition, it is basic to support the integration of distributed components independently of their implementation technology: the Common Object Request Broker Architecture (CORBA [6]) is the distributed object infrastructure adopted most often to put together complex C/S systems by simply assembling and extending (possibly legacy) components.

Mobile agents and CORBA are often perceived as two alternative technologies for the design, implementation and deployment of distributed services. CORBA permits the C/S interworking of heterogeneous service components, transparently to their location. Mobile agents suggest the implementation of services according to a peer-to-peer model of interaction and enhance the C/S programming paradigm by giving service designers the possibility to move location-aware executing entities [3-4]. We claim that not only MA and CORBA can coexist in the realization of the same service infrastructure but also that their integration simplifies the deployment of flexible services. On the one hand, the MA technology permits to distribute dynamically a set of agents to perform service-specific operations on data flows at both end-systems and intermediate hosts on the C/S network path. On the other hand, CORBA can enhance the capability of mobile
agents to be accessed by any CORBA client and to invoke any legacy functionality via standard interfaces.

According to these directions, the paper proposes a methodological approach to the integration of CORBA-based DMAs with MA infrastructures to ensure a high level of accessibility, customizability and adaptability to run-time determined system conditions. This effort has resulted in the design and implementation of MADAMA* (Mobile Agent-based Distributed Architecture for Multimedia Applications), that integrates a distributed CORBA-based DMA (JC-DiVA) with an open and secure MA programming framework (SOMA [7]). MADAMA is accessible either via CORBA client Web applets or via proprietary clients that can be dynamically installed by mobile agents on the end-systems that request a multimedia stream. In addition, MADAMA uses mobile agents to distribute service intelligence and control on network paths in order to realize an application-level monitoring of the QoS currently available for multimedia flows; distributed agents exploit the monitoring information either to provide feedback to the MADAMA servers or to perform QoS adaptation operations on flows. The paper reports the lessons learned from the implementation of the MADAMA prototype and the first performance results to show the feasibility of the approach.

2 Mobile Code for Service Flexibility

Several research activities have recently focused on the provision of service flexibility, intended as the property of network services of being accessible, customizable, extensible and adaptive [1-3]. The accessibility property requires services to provide standard and ubiquitously available interfaces, e.g., applet clients accessed via standard Web browsers. The customizability requires to collect user requirements and preferences, even specified and changed at run-time, and to consequently accommodate service provision. The extensibility usually indicates the capability of servers to either add or modify dynamically service protocols, components and features without suspending service provision. Adaptability identifies the possibility of dynamically reconfigure the provided service depending on observed modifications in the environment of execution, e.g., video servers can adjust the quality of transmitted streams in case of changes of the available bandwidth on the network path from servers to clients.

The paper claims that mobile code programming paradigms permit to significantly enhance the traditional C/S interaction model and are particularly suitable for the design, implementation and deployment of flexible services. Code mobility facilitates the extension and tailoring of servers via add-on modules that can be dynamically loaded to implement requested additional features. In addition, mobile code frameworks permits to implement services in terms of interworking active entities that can be dynamically distributed by need. In that way, services can be provided not only by end-systems, but also by any intermediate host in the network [1].

Among the paradigms proposed for code mobility [4,8], the paper focuses on the advantages of the adoption of the MA technology to implement a service middleware, capable of integrating with CORBA-based systems and of achieving service flexibility in the DMA area. Mobile agents not only permit dynamic distribution and upgrading of computing environments by need, without imposing service suspension, but also makes possible to move whole running entities (both code and execution state), thus preserving the operation results collected before agent migration. Services can greatly benefit from the implementation on top of an MA infrastructure to facilitate the achievement of the following properties:

- **High accessibility.** Agent migration simplifies the movement of any service component at run-time and in particular also of service clients. Agent-encapsulated proprietary clients can be dynamically installed at any requesting service access point, thus enhancing service accessibility even if it is compulsory to employ non-standard access interfaces.
- **Customizability.** Mobile agents are a technology suitable to support service customization depending on user preferences. For instance, dedicated agents can be in charge of collecting user profile information, of migrating profiles by need depending of most frequently access points, and of tailoring consequently all multimedia streams directed to that user, e.g., via distributed filtering and format conversion.
- **Distribution of service control.** Agents can provide and control MA-based services by moving at run-time to benefit from locality in resource control. In addition, agents can exploit even complex coordination mechanisms and models, e.g., based on shared tuple spaces [9]. For instance, agents can coordinate themselves at any time to update and re-negotiate the QoS level offered by an MA-based service, or to accommodate dynamically the entering/leaving of service users and providers.
- **Adaptability.** Agents make possible to adapt services depending on current system conditions and locality considerations (location-aware MA-based services [3]). For instance, monitoring agents can obtain a global/local view of the system state and MA-based services can exploit this information to perform corrective operations (e.g., request of additional communication channels). The MA technology also simplifies the fast prototyping of new protocols by permitting the dynamic distribution of network-aware agents that implement service-specific protocols only on interested

---

* MADAMA is available at http://lia.deis.unibo.it/Software/MADAMA/*.
network paths. In that way, mobile agents can accelerate the transition toward the acceptance and hardware implementation of new standards.

- **Security.** MA platforms not only integrate widely diffused mechanisms and policies for network security, but also provide specific solutions to the security problems that stem from the movement and execution of untrusted code.

- **Interoperability.** Agents work in an open global environment and should interoperate with existing resource/service components, from legacy systems to standard Internet services such as DNS. This degree of interoperability requires the MA system compliance with interoperability standards, such as CORBA, MASIF (the OMG standard for heterogeneous MA platforms [10]) and FIPA (the most diffused specification in the area of agent communication languages [11]).

An MA framework that provides the above properties is a very suitable middleware to implement flexible DMAs, especially if combined with the possibility of integrating with the CORBA distributed infrastructure, as described in the following.

### 3 The Integration of MA Infrastructures and CORBA-based DMAs

The section describes how some DMA features can be significantly enhanced via an approach that integrates CORBA-based interoperable infrastructures with MA programming frameworks.

#### 3.1 Accessibility

Accessibility identifies the possibility of users to access subscribed services independently of their current access point and of the capabilities of the involved end-systems.

CORBA provides basic mechanisms to enhance service accessibility. The encapsulation of service components into interfaces, described in a standard format such as the Interface Definition Language [6], permits the C/S interoperation between distributed objects. For instance, any client (whichever position it is running and whichever language it is implemented) can invoke a CORBA server object in a completely transparent way with regard to the server location and its implementation language. In addition, commercial CORBA implementations enhance these basic accessibility mechanisms with other features to simplify the integration in the Web via client applets.

Mobile agents can significantly complement the CORBA technology by simplifying the automatic installation of proprietary clients, disseminated by the server to the requesting service clients. Client installation is often a complex task that requires to understand thoroughly which software is already present on the target host, e.g., which operating system, and to choose the compatible version according to this information and to other system characteristics, such as the type of network the host is connected to. In addition, installation is a critical operation from the security point of view because it usually requires to write on file systems and to modify system properties. Mobile agents are very suitable for installation duties because agents can modify their behaviors depending on the dynamically determined characteristics of the target system and can operate locally to it after the needed security checks.

#### 3.2 QoS Adaptation

QoS adaptation, i.e., the possibility to change dynamically the characteristics (e.g., format, reliability, bandwidth requirements) of the transferred data flows depending on current properties of end-systems and traversed network paths, is emerging as a fundamental property of flexible services. In particular, QoS adaptation is critical when dealing with DMAs, because this class of applications usually requests the usage of a large amount of system resources and imposes strict constraints on timely data delivery.

Two main directions are emerging in the QoS adaptation area. Some work has defined and standardized new protocols to reserve network resources, to ensure their availability to the requesting services. The most notable example of this approach (indicated as hard QoS in the following) is the Resource ReserVation Protocol (RSVP) [12]. Other work is instead at the application level, and proposes infrastructures that enhance service provision with the possibility to require a specified QoS, to try to satisfy these requirements and to notify the service infrastructure in case of change in the currently available quality. All is done at the application level, with no strict guarantee of requirement satisfaction, but with no need to modify the underlying network layer, thus preserving the best-effort approach of the world-wide diffused Internet protocol. We define this approach soft QoS.

Hard QoS is the only way to ensure the real reservation and consequently the availability of the needed amount of network resources. While the provision of hard QoS guarantees is the objective of many DMA architectures, it is recognized that the soft QoS approach is a very interesting short-term solution to increase the quality of multimedia distribution in the current Internet infrastructure.

Flexible DMA services can exploit an MA-based application-level infrastructure for QoS adaptation (see the MADAMA case study and, in particular, Section 4.4) where mobile agents are dynamically distributed along the path between source and targets of multimedia flows.
to obtain a consistent view of the available resources in the distributed system, to be informed of any QoS degradation and to adapt to it by either notifying the DMA servers or locally performing transformation (e.g., data filtering, compression, transcoding) on multimedia flows [13-15]. Mobile agents are a very suitable technology for QoS monitoring and adaptation because they permit to distribute application-specific (possibly user-specific) service intelligence only at run-time, where needed and for the limited duration of service provision.

4 The MADAMA Case Study

MADAMA puts together the work we have accomplished in the area of CORBA-based DMA (JC-DiVA) and of MA programming environments (SOMA), and represents an interesting case study to check the effectiveness of our integrated approach in the realization of flexible services.

4.1 The SOMA Programming Environment

SOMA* (Secure and Open Mobile Agents) [16] is a Java-based programming environment that provides a rich infrastructure of services to fasten and simplify the design, implementation and deployment of MA-based applications. Mobile agents are location-aware entities and SOMA provides a hierarchy of locality abstractions suitable for describing any kind of interworking scenario: any network node has one place for agent execution; several places are grouped into domain abstractions. In each domain, a default place hosts a gateway abstraction that is in charge of inter-domain coordination routing and coordination. Locality abstractions have a direct mapping to physical resources: agents execute in places that represent physical nodes; domains are suitable for modeling physical LANs. For instance, in the case of the implementation of a MA-based multicast infrastructure, the visibility of allocation makes possible to recognize whether there are several receivers within the same domain locality to minimize the bandwidth usage by duplicating packets only when needed.

In addition to these locality abstractions, SOMA provides a rich set of MA-based services to simplify the design, implementation and deployment of flexible services. The layered and modular architecture of services in SOMA is depicted in Figure 1, and can be dynamically extended to answer new user/service-specific requirements. A full description of SOMA services is presented elsewhere and is out of the scope of this paper, where we give some details only on the SOMA services for interoperability and QoS that are extensively exploited in the implementation of MADAMA.

The SOMA interoperability service permits agents to integrate with different resources and service components, whether MA-based or not. The service is based on compliance with CORBA [6]: the SOMA framework provides a CORBA facility to simplify the implementation of agents that exploit other CORBA components (SOMA agents as CORBA clients) and that offer their services to external systems (SOMA agents as CORBA servers). In addition, SOMA is conformant to the Mobile Agent Facility Specification (MASIF) [10] that standardizes functions for agent management and transfer by using CORBA as the standard bridge between heterogeneous MA platforms.

The SOMA QoS service is in charge of application-level QoS monitoring and adaptation functionality. The monitoring module can observe the state of local resources and services and provides this information to the application layer. SOMA agents can monitor both system indicators (e.g., CPU load, file system occupation, printer status, available network bandwidth and collision rate) and application indicators (e.g., available services, program versioning, local agent states). They exploit platform-dependent mechanisms to obtain the monitored indicators (e.g., the psapi.dll on Windows NT and psstat on Unix); they invoke the functionality provided by the hosting operating system via the Java Native Interface [17]. SOMA-based flexible services can either access directly the monitored properties or refine general-purpose SOMA agents for QoS Adaptation (QoS Negotiator agents in Section 4.4). Depending on the monitored information, service components or the SOMA infrastructure can enforce a strategy suitable for adapting to the current environment conditions, without suspending service provision.

![Figure 1. The SOMA service architecture](image)

---

* SOMA is available at [http://lia.deis.unibo.it/Research/SOMA/](http://lia.deis.unibo.it/Research/SOMA/).
4.2 JC-DiVA

JC-DiVA\(^\text{1}\) (Java/CORBA Distributed Video Architecture) is a distributed DMA prototype: it provides an interactive C/S service for retrieval and real-time delivery of video sources from a remote distributed database.

The overall architecture of JC-DiVA is depicted in Figure 2. A set of Film Servers (FS) manages a distributed data base of media resources, namely the video movies. An Archive Manager Server (AMS) is responsible to provide to the JC-DiVA client the functionality of listing the movies available in the distributed data base, and to provide the client with the URL of a Film Server which stores the selected resource. The AMS stores and periodically updates a data structure containing the list of available media resources, each linked with the corresponding film server. The AMS provides also a database management interface for a proper system administration client.

After locating the right media, the user application (i.e. the JC-DiVA client) binds directly to the appropriate film server for the streaming of the movie, using the most suited transport protocol for streaming over the networks. For this purpose, a streaming session is established directly between the client and the FS. Currently, RTP streaming is supported [18]. The client implementation is based on the Java Media Framework, an application programmer interface for incorporating time-based media into Java applications and applets [19].

The system offers two categories of media delivery services, asynchronous and synchronous. The transmission of multimedia flows over the network in the asynchronous service is based on the Real-Time Control Protocol (RTCP). RTCP is a part of RTP and helps with synchronization and QoS management. With this approach a different "tunnel" of RTP packets is created between the front-end server and the client. The synchronous service allows synchronized video delivery to a number of clients and exploits, when supported, IP multicasting functionality.

All JC-DiVA components are CORBA objects that interact by means of the ORB middleware bus. Figure 3 shows the component diagram of the JC-DiVA access system.

![Figure 2. Architecture of the JC-DiVA system](image)

![Figure 3. UML component diagram of the JC-DiVA access system](image)

However, though the connection between the CORBA client and the MADAMA service is transparently maintained by the CORBA infrastructure, the client applet cannot take advantage of the possibly available infrastructure of intermediate active hosts presented in Section 3. For this reason, we have also implemented a proprietary client of MADAMA that directly accesses communication channels to exploit the MA-based QoS adaptation functionality presented in Section 4.4. The proprietary client has enhanced functionality with regards to the applet one: it allows users to specify the required QoS level and to modify requests during service provision, and is able to connect to the dynamically determined chain of

---

\(^\text{1}\) JC-DiVA is available at [http://grid.grid.unina.it/projects/diva/](http://grid.grid.unina.it/projects/diva/).
intermediate hosts along the path from the client to the selected server.

To permit the usage of the proprietary client also from hosts that have not it already installed, we have exploited the SOMA framework to make possible its dynamic installation (see Figure 4). Users can request the installation via a Web form; the request triggers the migration of an installation agent that carries the MADAMA client from a code server to the client host.

When the multimedia flow of the selected film has terminated, the agent asks the user either to cancel the installed files or to maintain the full client for future sessions of the MADAMA service.

Figure 4. Web request for the MA-based installation of the MADAMA client (a) and its GUI (b)

The installation of the full client via mobile agents has different costs depending on several factors, from network distance and available bandwidth between client hosts and code servers, to the possible overhead imposed by security requirements. We have measured the time to install the proprietary client (85kB) on 300-MHz PentiumII PCs while the code server is a 120-MHz Sun SparcStation5; the hosts are located in two different 10-Mb Ethernet LANs. In case of trusted environment, i.e. if the user can trusts the installation agent and does not require any security check, the average required time is 3855 ms (3487 ms if the HotSpot just-in-time compiler is active on the receiving host [20]). In case of untrusted environments, the installing agent is first authenticated, its code integrity is verified, and its access to the client host disk space is controlled on the basis of the local security policy. In this case, the average required time goes up to 4492 ms (3912 ms with HotSpot running).

4.4 QoS Adaptation

To enhance the JC-DiVA capacity, stemming from the RTP adoption, of controlling end-to-end time constraints on multimedia flows, MADAMA integrates an MA-based infrastructure for the fast prototyping of QoS adaptation protocols. This infrastructure is dynamically distributed along the paths between source and targets of multimedia streams; it permits users to request different QoS levels for the various transmitted streams; finally, it allows to manage and adjust the requested quality during service provision, to respond to user addition/deletion from a multicast group and to modifications of resource availability.

The infrastructure is composed by two different types of coordinated agents: the QoS Negotiators (QoSNs) and the Admission Controllers (ACs) (see Figure 5).

Figure 5. Tunneling, co-routing and multicast in MADAMA

ACs have been realized by tailoring SOMA agents for QoS monitoring. ACs are present on every node of the network; this assumption is not severe because they are implemented by mobile agents that can move and be installed whenever they are needed. Each AC manages local resources and keeps track of their current commitment to already accepted streams. The flow specifications of streams are recorded in local <receiving-host, bandwidth, delay, loss> tuples [21]. Any tuple represents the statistics of MADAMA traffic between the local and the receiving host: the first time, it contains values calculated upon a short sample of communication; then, it is updated by monitoring real traffic of current MADAMA sessions. ACs are in charge of answering to reservation requests from QoSNs.

QoSNs agents are dynamically distributed at the JC-DiVA FS (source), at JC-DiVA clients (targets) and at some intermediate nodes. They maintain session state by storing user preferences and flow specifications for any multimedia stream. QoSNs evaluate the feasibility of meeting these requirements against the local AC database and exploit the SOMA communication service to perform the negotiation phase for the definition of the achievable QoS.

Let us first consider the simple scenario of single stream addressed to single target. The path between source and target is automatically determined at run-time, by tracing the route of a dummy packet. QoSNs move to the chosen hosts on the path and query the AC database. If not enough resources are available for the desired QoS, QoSNs agree on reduced requests and notify this to the JC-DiVA FS. The FS is able to instantiate more types of
Several research works have recently explored the idea of realizing dynamic and adaptive infrastructures for multimedia flow distribution. Many proposals come from the active network research area: active networks propose the programmability at the network layer with two possible approaches: the *integrated* one, where any exchanged message contains its handling code, and the *discrete* one, where the new behavior is out-of-band forwarded to active nodes. The integrated approach is mainly used to reserve resources on the path between the DMA server and its clients and to dynamically inject application-specific multicast protocols, while most of the proposals pertain to the discrete approach area, where intermediate hosts play the role of active filters that dynamically adapt the multimedia service to currently available bandwidth [23].

Some researchers argue that active networks are not the most suitable solution for the implementation of network-centric services, because of their proposal at the network layer; by working at the application level, they prefer to concentrate on the provision of more effective and programmable service architectures [13]. Among the proposals at the application-level, some works are emerging in the field of multi-agent systems, mainly focused on the establishing and maintaining as far as possible a user-defined level of QoS [14]. The multi-agent approach is similar to ours from many points of view, but multi-agent service components suffer from a lower degree of flexibility, since they do not have the possibility to migrate during service provision and are required to be present on the interested hosts before the starting of the multimedia flow. Finally, it is recognized that the MA technology is intrinsically suitable for this kind of applications, because of its application-level approach that maintains location awareness for the MA-based applications, and obviously because of the flexibility given by the agent mobility [5,24]. To our knowledge, however, there are not implemented DMA prototypes yet that use mobile agents to enhance the flexibility of multimedia flow distribution.

6 Conclusions

We believe that distributed objects and mobile agents technologies may complement each other in the provision of innovative solutions by answering to the issues of dynamicity, extensibility and flexibility in distributed services. The paper proposes the architecture of a prototype system that aims at integrating the CORBA C/S middleware and SOMA mobile agents to provide some advanced features for DMA services.

We have described the proposed integration approach for a typical DMA, namely the JC-DiVA service. The MADAMA case study shows the benefits of integration in providing flexibility in terms of enlarged accessibility and re-configuration in response to changing resource availability. On the one hand, the availability of the

5 Related Work

codes (the codec is the object in charge of converting each track of media data from one input format to another) [19]. This means that the same media data (e.g., a particular lesson or film) is stored onto the multimedia service providers with different formats (e.g. MPEG, H.261, H.262, etc). Thus, the QoSN transmit the QoS level indication to the FS. The FS then selects the most appropriate content type (for example an AVI file, which surely support the H.263 format -medium quality-, with a low bandwidth requirement). If a client is located in a subnetwork which has a high speed connection with the multimedia service provider, or if the client has established, by means of the interaction with the QoSN agents, that there are enough resources for the requested QoS, then the FS instantiates the codec with the best quality, for example MPEG content type with MPEG-1 format. After a successful negotiation phase, the (possibly scaled) multimedia stream starts to flow.

In the case of multicast distribution of the same multimedia stream (for N targets), the generated network traffic can be limited by exploiting location awareness of agents. QoSNs can ascertain whether there are several targets within the same domain locality and can split packets only when it is necessary. This is usually commanded by the QoSN at the gateway of the last domain (see Figure 5).

MADAMA requires the presence of the above described infrastructure composed by distributed and coordinated agents. After the setup phase has negotiated the service levels, the stream can flow from source to target. During service provision, the flow can also be dynamically adapted, to adjust the required QoS level at runtime with a best-effort approach. Dynamic adaptations introduce overhead, that is only a percentage of the one required by the setup phase. For that reason, we report about the setup costs to instantiate and send ACs (about 6KB-sized) and QoSNs (about 4KB-sized) in parallel to interested nodes. We have considered the worst case when none of the intermediate nodes has neither the AC nor the QoS agent; more commonly, hosts may have already the AC agent running for purposes of remote monitoring and diagnosis. Performance has been measured on two 10-Mb Ethernet LANs of heterogeneous hosts (PentiumPCs with Windows NT 4.0 and Sun SPARCstation with Solaris 2.5) in the case of a FS 5 hop far from its targets, i.e. the multimedia stream has to pass through 4 intermediate non-tunneled nodes to reach targets. In the case of untrusted environments, the average setup time is 5907 ms (5221 ms with HotSpot); in the case of trusted environments, MADAMA gives the possibility to disable security checks on incoming agents, with a considerably reduced average setup time (4135/3394 ms without/with just-in-time optimizations).
SOMA infrastructure has permitted to dynamically create and distribute mobile entities. On the other hand, compliance with CORBA has allowed the interoperation with most common standard components and the usage of well-spread interfaces, such as the Web-based ones. Apart from the general feasibility considerations, the first MADAMA prototype has exhibited acceptable performance results both for the MA-based installation of the proprietary client and for the setup costs of the MA-based QoS infrastructure.

After these first results, we are focusing on the extension of MADAMA along two main guidelines: load balancing, to exploit both the CORBA replication service and the agent mobility in the dynamic distribution of DMA requests among MADAMA servers based on their computing capabilities and on the current network traffic; conformance with the OMG Multimedia Streaming Framework [25] to permit the CORBA-based interoperability between MADAMA and other DMAs implemented by university partners within the Italian MOSAICO project.

Acknowledgements
Work carried out under the financial support of the Italian Ministero dell’Università e della Ricerca Scientifica e Tecnologica (MURST) in the framework of the Project “MOSAICO: Design Methodologies and Tools of High Performance System for Distributed Applications” and the Italian “Consiglio Nazionale delle Ricerche” in the framework of the Project “Global Applications in the Internet Area: Models and Programming Environments”.

References