An Integrated Management Environment for Network Resources and Services

Paolo Bellavista, *Student Member, IEEE*, Antonio Corradi, *Member, IEEE*, and Cesare Stefanelli, *Member, IEEE*

***Abstract*—Technological and human factors have contributed to increase the complexity of the network management problem. Heterogeneity and globalization of network resources, on one hand, have increased user expectations for flexible and easy-to-use environments; on the other hand, they have suggested entirely novel ways to face the management problem. Several research efforts recognize the need for integrated solutions to manage both network resources and services in open, global, and untrusted environments. In addition, these solutions should permit the coexistence of different management models and should interoperate with legacy systems. In the paper, we define a general architecture based on a distributed processing environment (DPE) that offers a large set of facilities to the application level. We have developed the MESIS management environment shaped after the above architecture and its DPE facilities with the mobile agents technology. MESIS handles, in a uniform way, both resources and services, and focuses on two crucial properties: interoperability to overcome heterogeneity, and security to grant users safe and protected operations. The Agent Interoperability Facility supports compliance with CORBA-based management systems and with MASIF agent platforms. The Agent Security Facility provides authentication, integrity, privacy, authorization, and secure interoperation with CORBA systems.**

***Index Terms*—Distributed systems, Internet services, interoperability, network management, security.**

I. Introduction

RECENT directions in the evolution of network technology, such as the global scale of interconnection and the availability of high-speed broad-band networks, have forced us to consider network resources as components of a global distributed system. In addition, the deregulation of the telecommunications industry, and its convergence in goals with the area of distributed information systems, request openness to achieve interoperability among resources, tools, and services. The Internet, which is likely to be pervasive in the evolution of telecommunications, is the best example of the new network scenario of open and global distributed systems, where solutions should be scalable to face globality and should provide interoperability to cope with heterogeneity of network components.

Manuscript received March 17, 1999; revised October 25, 1999. This work was supported by the Italian Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST) under Project "MOSAICO: Design Methodologies and Tools of High Performance Systems for Distributed Applications."

P. Bellavista and A. Corradi are with the Dipartimento Elettronica, Informatica e Sistemistica, Università di Bologna, 40136 Bologna, Italy (e-mail: pbellavista@deis.unibo.it; acorradi@deis.unibo.it).

C. Stefanelli is with the Dipartimento di Ingegneria, Università di Ferrara, Ferrara, Italy (e-mail: cstefanelli@ing.unife.it).

Publisher Item Identifier S 0733-8716(00)02766-9.

The evolution of the network scenario has suggested that we consider new management models to overcome the limits of traditional centralized client/server approaches. There is a growing interest in taking into account Web-based management systems [1], [2] and in adopting integration standards such as CORBA that also permit us to deal with legacy components [3]–[6]. There is a strong emphasis on the use of mobile entities to provide flexible, scalable, and effective management solutions by programming network resources dynamically [7]–[14]. There are also encompassing efforts in defining open architectures to integrate the management of traditional telecommunications with new distributed services [15], [16]. Recent research approaches recognize the following important issues in resource and service management for open, global, and untrusted systems:

- to facilitate delegation and automation of control actions, thus reducing network load, relieving the central manager duties, and improving scalability;
- to address the management of heterogeneous network elements by focusing on interoperability and by promoting acceptance of new standards;
- to help in the design and fast deployment of new services, improving user customization and avoiding time-consuming redesign;
- to provide secure environments on top of intrinsically untrusted networks.

New management approaches propose solutions to the above issues with different peculiarities and at different levels of abstraction. For instance, consider the case of resource allocation that can be either visible or transparent to managers. While allocation visibility permits us to obtain efficient solutions, allocation transparency helps us to cope with the complexity of internetworked systems. Active Networks (AN's) exemplify how allocation visibility can be used for management purposes and also for introducing new protocols without discontinuing system operations. On the contrary, CORBA-based solutions propose a higher-level approach that hides allocation to applications, simplifying the development of distributed services. We believe that a management environment should offer both allocation visibility and allocation transparency. The former is compulsory to express management policies and to obtain efficient solutions, while the latter is preferred by final users and when designing complex distributed services.

We feel that the main issue still to be faced is the definition of a comprehensive solution for the integrated management of both network resources and services, able to provide all the features required by different levels of usage and with different levels of abstraction [17].

The paper presents the organization of an integrated management environment, and proposes a general layered architecture. Management applications exploit an underlying distributed processing environment (DPE) facility layer, which should provide all the functionality needed when dealing with open, global, and untrusted environments. Many different DPE implementations may coexist and interoperate in providing DPE facilities.

Following the guideline of the proposed architecture, we have designed a management environment, called MESIS (Management Environment for Secure and Interoperable Services—`http://lia.deis.unibo.it/Research/MESIS/`), and implemented with a Mobile Agent (MA) technology [18]. MESIS DPE facilities are implemented in terms of mobile agents to facilitate delegation and management automation, and to achieve efficiency and scalability through local access to managed resources. In particular, the paper focuses on the Agent Interoperability Facility (AIF) and the Agent Security Facility (ASF). The AIF permits us to interoperate with already existing services and legacy components, via compliance with OMG CORBA [3]. In consideration of the increasing diffusion of MA systems, AIF is also conformant with the Mobile Agent System Interoperability Facility (MASIF) [19] to permit agent exchange between MA platforms. The ASF answers the typical security concerns of management in the Internet environment, which is global, open, and untrusted by nature. It provides authentication of principals, integrity, privacy, authorization, and accountability when accessing to resources.

The paper is structured as follows. Section II provides an overview of novel solutions emerging in network and systems management. Section III motivates the need for a layered approach to face the raising complexity of management and presents our proposed architecture. Section IV illustrates the design and the functionality of MESIS and focuses on the facilities for interoperability and security. Section V explains how MESIS can be employed to manage complex network services, such as in the Video on Demand application domain. Concluding remarks follow.

II. Novel Management Solutions

Recent research has proposed several different approaches to overcome the limits of traditional management systems. We do not want to give a general overview of these approaches, but only try to sketch their peculiarities and to identify their main differences. In particular, we stress that proposed solutions are at different levels of abstraction, and suit different specific issues in the management domain.

The main idea in AN's is to program network components, so that users can directly modify the behavior of the network itself while it continues to operate. AN's push programmability down to the network layer of the OSI protocol stack, and have already shown their capacity to achieve significant results in terms of flexibility, performance, scalability, and QoS provision [13], [20], [21]. However, there are several typical management issues which are difficult to solve at the network layer. For instance, there is no general agreement on the level at which security should be faced, and people discuss whether security should be considered at either the network or the application layer [22]. We believe that many security-related tasks are more easily addressed at a higher level. For instance, user authentication requires public key infrastructures usually available as application level tools, and also the association of authenticated users with recognized roles needs application level facilities for defining and managing the proper trust model [23].

Other solutions that make use of code mobility for network management come from the MA research activity [9]–[12], [24], [25]. A mobile agent is a program that acts on behalf of a user and is capable of moving autonomously within the network. Probably the most limiting feature of MA-based management approaches seems to be the fact that only a few MA platforms address interoperability with existing management components, whether MA-based or traditional ones [27], [28]. In addition, they do not generally provide a layered architecture of common services, making the development of complex management applications difficult.

CORBA is the most widely diffused architecture to deal with distributed heterogeneous programming. However, even if CORBA has raised great interest in the management area, it currently seems more to play the role of integration technology among existing solutions (CORBA gateways toward SNMP/CMIP components [5], [6]) than to propose a framework to build new CORBA-based management applications. Some peculiarities of CORBA partially limit its use in the management area: CORBA-based applications are typically location unaware, while managing distributed resources and services usually request visibility of topology and locality information. In addition, CORBA implementations lack abstractions for managing object groups, even if the collection abstraction is clearly necessary for the management of replicated services [29], [30]. Finally, the interaction of objects using diverse security technologies is complex because CORBA does not standardize the possibility to negotiate security technology [31].

Other proposals are abstracted from implementation technologies and describe solution frameworks at the architecture level. The Telecommunications Management Network (TMN) framework [16] goes beyond the manager/agent model of OSI systems management [32] by introducing a distributed set of cooperating systems for monitoring and control, conceptually separated from the telecommunications network being managed [17], [21]. TMN's main limitation for highly dynamic and open systems seems to be its client/server management model.

The Telecommunication Information Networking Architecture (TINA) [15] proposes a solution at a higher level of abstraction. TINA architecture is directed to design any kind of service, running on a global scale and on different network technologies. TINA suggests a uniform support for management where the management itself is considered a service. TINA applications, service components, and network resources reside on top of a DPE, which can hide the complexity of distribution and heterogeneity from service developers. Unfortunately, to present a global solution, TINA seems to push toward very complex implementation, so that some research work has addressed the issue of implementing simplified architectures of TINA to offer an earlier opportunity for cost-effective evolution of current networks [33].

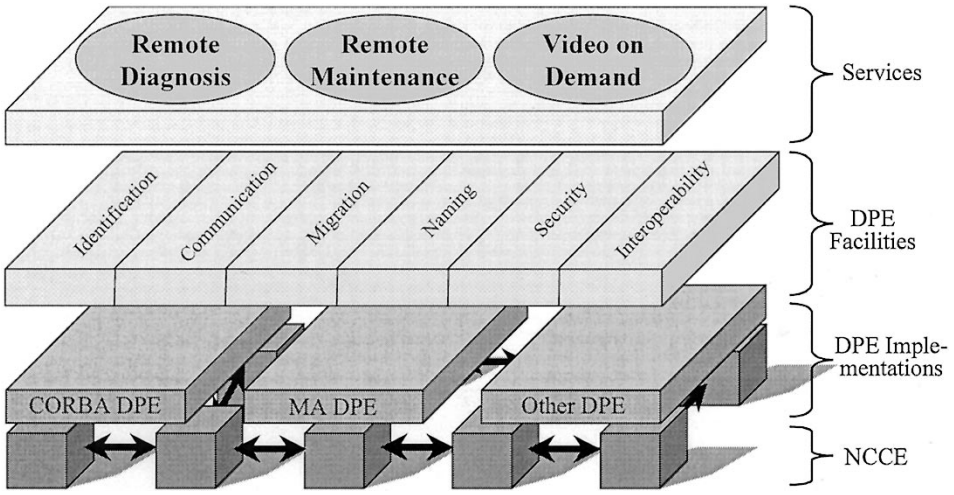

Fig. 1. Our architecture for the design of management environments.

III. A Layered Architecture for Network Management

Modern management environments should handle both resources and services. Resources are not only heterogeneous network components, such as routers, hosts, and LAN's, but also logical abstractions, such as file systems and processes. Services are distributed applications, and have recently evolved from simple ones, such as ftp and telnet, to more advanced ones, such as remote software configuration and teleconferencing.

Novel management approaches should provide integrated solutions that permit us, in a uniform way, to design, deploy, and control user-oriented services, and to manage logical and physical resources. In addition, they should provide the widest set of the following properties to face the new challenges of globally interconnected networks.

1) Flexibility: New protocols and services should be introduced dynamically, to answer to application-, session-, or user-specific requirements; for instance, a customized multicast protocol can be deployed at runtime to support a teleconference service.

2) Adaptability: Services should adapt to current network situation and should evolve with user requirements with no need to suspend during their phase of tuning; for instance, a specific level of QoS can be renegotiated when congestion no longer makes possible its provision.

3) Interoperability: Services should take advantage of any other existing service and resource, from diffused management components (based on SNMP and CMIP) to management legacy systems, from CORBA-compliant services to any other resource that exploits possibly different emerging standards.

4) Distribution of Service Control: In general, services are not furnished by one predefined service provider, but by several providers allocated anywhere in the system. This motivates the need for distributing replicated service controllers to avoid the bottleneck of centralized management; for instance, a distributed control could be more efficient for a geographically distributed teleconference service.

5) Coordination of Services and Service Providers: Any service can coordinate with other ones to negotiate any required strategy, and different providers can cooperate to offer coordinated services; for instance, only the coordination between network operators and multimedia providers can guarantee a specified bandwidth and latency in a video-on-demand service.

6) Security: Services are distributed in a global and untrusted environment, shared among a multitude of users and providers, in a scenario that imposes strong requirements for security; for instance, a specific service should be available only to recognized users, and any service should choose the proper balance between efficiency and required security level.

We have defined an architecture that should simplify the realization of an environment, with the above properties, for the integrated management of both resources and services. The architecture is organized in layers, as depicted in Fig. 1. Services at the application layer are implemented by exploiting the underlying layer of DPE facilities that we consider fundamental for the realization of any distributed application in open and untrusted environments.

DPE facilities are supported by the underlying DPE implementations, but should not rely on a specific one to provide flexible solutions. For instance, a naming facility can be built on different naming systems provided by different DPE implementations, e.g., DNS-, CORBA-, and LDAP-compliant naming services [34], [30], [35]. Our architecture suggests several DPE implementations to coexist; any DPE, in its turn, abstracts and hides specific details of the underlying native computing and communication environment (NCCE).

With regard to the DPE facilities layer, let us note that each facility answers specific problems, and can also interact with other facilities, e.g., naming and security facilities interact when the system has to authenticate an entity and to recognize its role [23]. Our architecture recognizes the following set of facilities.

The *identification* facility permits us to tag resources, users, and services by assigning unique names to entities in the system.

The *migration* facility is in charge of transporting one entity that should change its allocation from its sending node to the destination one. The reallocated entity, if it is active, should transparently restart its execution at the new location.

The *communication* facility supports any exchange of information between service components, and is capable of eventually delivering messages to reallocated entities.

The *naming* facility accommodates and organizes all names of public entities and makes it possible to search and trace them, also in case of their migration.

The *interoperability* facility permits interoperation among different resources and different service components, designed with any programming style, by closely considering conformance with accepted standards.

The *security* facility protects any entity in the system, by providing a wide range of mechanisms and tools for authentication, authorization, controlled access to all resources and services, privacy, and integrity.

Although some management environments could neglect migration and the corresponding facility, we consider the possibility of reallocating entities as a basic feature when dealing with open and dynamic systems. Apart from the pioneer MbD work [7], both the MA and AN proposals adopt this perspective [11], [13].

IV. The MESIS Environment

The architecture presented in Section III has guided the design and the implementation of the MESIS integrated environment, created not only for managing distributed network resources, but also for supporting the easy definition, deployment, and tailoring of new network services. MESIS implementation is tied to the Mobile Agent paradigm (and to a corresponding MA DPE), because MA is a promising technology for dealing with the complexity of an open network-centric scenario. The MA support is written in Java and exploits the Java platform independence to face heterogeneity [36].

MESIS mobile agents fulfill administration needs by moving and executing on different nodes. Automation of control is obtained through the possibility of delegating management actions to agents who act autonomously and in a completely asynchronous fashion with respect to the administrator, thus relieving the agent's duty; for instance, one agent can automatically take care of software upgrading on dynamically selected nodes of a managed network. Mobile agents are permitted to adapt to system modifications by tuning the behavior of network resources and services at runtime; for instance, any administrator can modify and propagate security policies at any time, with no need to shut down the whole system, by dynamically instantiating new mobile agents to propagate the new policies in the administered domains.

A. The MESIS Architecture

MESIS has been designed with the goal of providing an integrated environment that addresses all the typical management issues of complex organizations. Organizations usually consist of several departments, even geographically distributed over the Internet. Each department has its private LAN, and needs to interact via gateways with other departments to accomplish coordinated tasks. In addition, the globality of the scenario addressed by MESIS forces us to face up to the issue of scalability.

For that reason, we consider it fundamental to introduce and to make possible the handling of the *locality* concept: MESIS embeds locality via a hierarchy of locality abstractions suitable for describing global distributed systems, ranging from simple LAN's to the Internet (see Fig. 2). Any node part of the MESIS environment hosts at least one *place* for agent execution and management; several places are grouped into *domain* abstractions. In each domain, a *default place* hosts a gateway which is in charge of interdomain routing functionality. The locality permits us also to introduce a scope when considering all other system policies, and helps in granting a protected framework for the belonging entities.

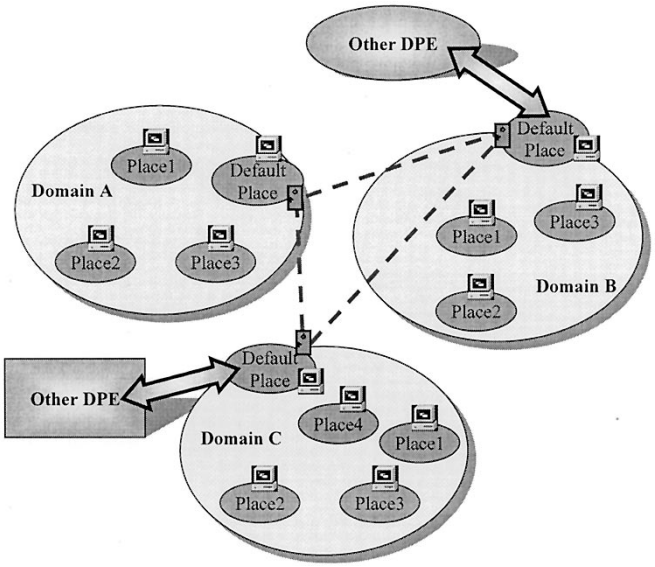

Fig. 2. MESIS locality abstractions.

The core part of the MESIS project is its architecture, which has been designed along the guidelines of Section III, and offers a distributed infrastructure with a set of facilities for the design and the development of complex network-centric applications (see Fig. 3). All facilities are implemented on top of the MESIS DPE layer by a set of coordinated mobile agents. In addition, the openness property of the MESIS infrastructure permits us to extend the programming framework by dynamically adding new services, even built on the already provided functionality.

MESIS DPE facilities are split in two levels: the lower one that groups the basic and primary mechanisms and the upper one that comprehends more evolved tools and services. The MESIS upper layer facilities (ULF) represent advanced operations and support directly the development of applications and services as follows.

1) *Agent Interoperability Facility (AIF)*: The AIF offers interfaces to simplify the calls from MESIS components (of both DPE and service layer) to external CORBA components or services; in addition, it supports the registration of MESIS-based services as CORBA servers; finally, it provides interoperability with different MA systems by implementing the MASIF standard interface (see Section IV-B).
2) *Agent Security Facility (ASF)*: The ASF provides all the mechanisms for authentication, authorization, integrity, and privacy (see Section IV-C). MESIS integrates a security framework based on standard security providers and

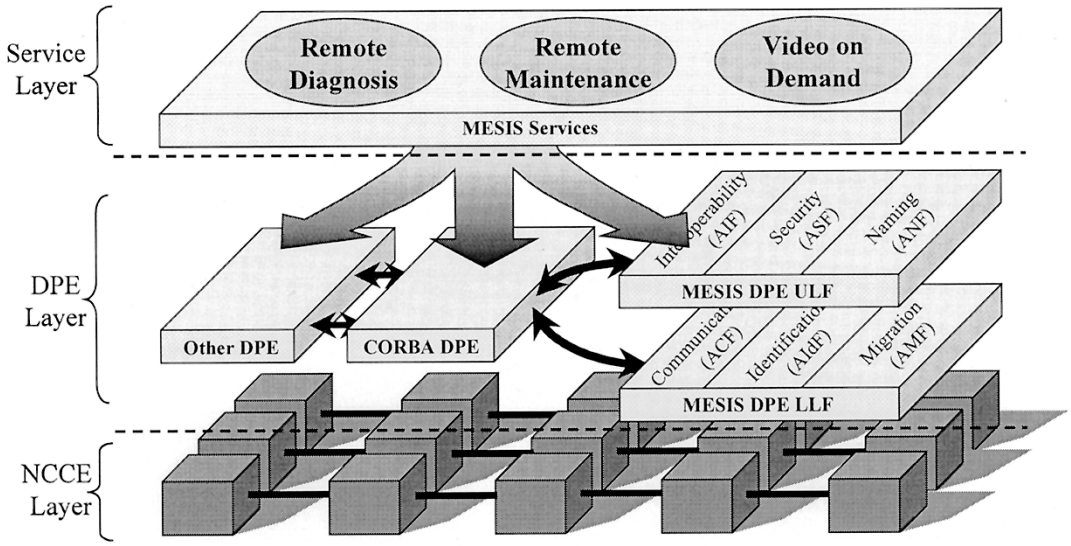

Fig. 3. MESIS architecture for management applications.

certificate infrastructures [37], [38]. The current ASF implementation is based on agents but can also interoperate with CORBA Security Services [30].

3) *Agent Naming Facility (ANF)*: The ANF dynamically maintains and permits access to the information about the current state of entities in the system (or in some of its parts). For example, it realizes a Domain Name Service and a Directory Service functionality. The ANF puts together a set of different naming systems, possibly characterized by different policies, and is implemented by a coordinated set of dedicated agents.

The AIF, ASF, and ANF can make use of the lower facilities in their implementation; for instance, the naming facility exploits the underlying identification facility. The MESIS Lower Layer Facilities (LLF's) include the following.

4) *Agent Migration Facility (AMF)*: The AMF gives service designers the possibility to simply reallocate network resources and service components at runtime. Entities capable of reallocation are represented by agents that can move in the network either via MA native migration methods or via standard specifications, such as CORBA Internet Inter-ORB Protocol [3] and MASIF [19].
5) *Agent Identification Facility (AIdF)*: The AIdF permits us to dynamically assign tags to any entity in the system. Globally unique identifiers are the basis for the realization of the multiple naming systems provided by the ANF that associates N different names with the same entity.
6) *Agent Communication Facility (ACF)*: The ACF provides mechanisms and tools to simplify coordination and communication between entities. Agents in the same place interact by means of shared objects, such as blackboards and tuple spaces. Any place hosts a Local Resource Manager module that regulates agent access to the node resources. This module controls the authorization of agents and enforces the place security policy. Whenever one agent needs to share one resource with another agent that resides in a remote place, it is forced to migrate to that place. Outside the scope of the place, agents can perform coordinated tasks by exchanging messages delivered to agents, even in case of migration.

The above facilities are available in different flavors, depending on system and service needs. For instance, the ANF, which exploits the underlying AIdF, currently permits the coexistence of our naming native service deriving from DNS with the CORBA Naming Service. Other LDAP-compliant naming and directory services are under integration to let users and designers choose among multiple name spaces. System- and application-specific considerations typically guide the selection of the available facilities to use; for this reason, a flexible management environment has to give service designers the possibility to choose the proper solution among a wide variety of available ones.

From the MA programming paradigm, MESIS inherits the introduction of the migration facility as a basic DPE functionality: it is intrinsic to the paradigm the reallocation of entities that move close to the locality enclosing the information to work upon. A DPE based on a different paradigm could choose different directions: a CORBA DPE would probably neglect this facility on the basis of allocation transparency. A DPE that implements all facilities, including the migration one, does not limit the flexibility and expressive capacity of a general management support.

In the following, we focus on the *Agent Interoperability Facility* and the *Agent Security Facility*, because the MESIS framework recognizes interoperability and security as basic requirements for any management environment. On one hand, any proposal of new network protocol and service should primarily consider the possibility of integration with legacy components and systems by respecting standard recommendations and solutions. On the other hand, the possibility of moving possibly untrusted pieces of code into network nodes answers the requirement for flexibility in service provision, but also obliges us to face the raised security problems: the environment should effectively and efficiently grant the proper security level in protecting resources from malicious intrusions.

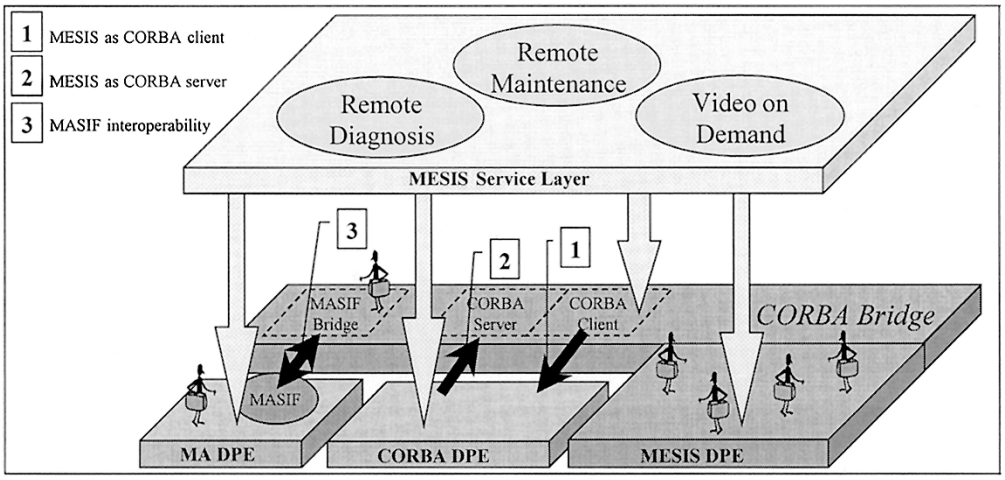

Fig. 4. MESIS interoperability via the CORBABridge add-on module.

B. The Agent Interoperability Facility

CORBA is the most widely accepted technology to overcome network, platform, and programming language heterogeneity and has acquired a central position in the evolution of telecommunications environments [15], [39], [40].

Within the MESIS DPE, the AIF module implements full compliance with CORBA. This compliance introduces some overhead, but the resulting openness and component stability to applications represent an invaluable saving of investments in the design of services. The AIF facility is implemented by the *CORBABridge* add-on that is composed by two modules: the *CORBA Client/Server* and the *MASIFBridge*. The AIF facility permits us to achieve interoperability along the following directions (see Fig. 4) [18].

5) A MESIS DPE component may call any CORBA DPE component or any service with a published CORBA interface by means of the *CORBA Client/Server* module (MESIS as CORBA clients).
6) Any service, whether MA-based or not, may exploit the *CORBA Client/Server* module to call MESIS DPE services, which publish CORBA standard interfaces (MESIS as CORBA servers).
7) MESIS DPE may interwork with any other MASIF-compliant MA DPE via the *MASIFBridge* (MASIF interoperability).

The first interoperability functionality allows MESIS agents to act as CORBA clients, for example, to control legacy network components via CORBA interfaces [5], [6], to exploit services and facilities provided by any CORBA DPE (e.g., Transactions, Collection, and Trader Object Services [30]), and to invoke CORBA-compliant application components [4].

MESIS DPE components can also become CORBA servers to offer their services to other entities. This enlarges the accessibility of a MESIS service to any existing client, independently of its technology (even non-MA). For instance, a user can employ a standard Web browser interface (such as a CORBA client applet integrated in a standard HTML page) to invoke the Video on Demand service presented in Section V.

The third interoperability direction derives from the increasing number of MA systems recently proposed and already implemented [41], [27], [28]. This number, on one hand, declares the interests in the MA paradigm; on the other hand, it can endanger interoperability and could limit the industrial diffusion of MA applications. MASIF proposes a standardization for agent and agent system names, and for agent system types and location. It defines two interfaces, `MAFAgentSystem` and `MAFFinder`, respectively, for agent management and for agent tracking. Agent management allows an external system to control agents belonging to a compliant MA platform: MASIF defines actions to suspend/resume/terminate agents and to move agents between MA systems provided that they have compatible agent types. Agent tracking permits us to locate agents and their services. Agents are registered with `MAFFinders` that provide an MA global naming service more suitable than the CORBA one for entities mobile by nature, like agents.

The implementation of the CORBABridge is based on VisiBroker [42]. However, the module is portable without modification on any ORB compliant to the CORBA 2.2 specification. In fact, we have used only the portable functions provided by the Internet Inter-ORB Protocol and the Portable Object Adapter [3], to overcome potential incompatibility among different ORB's.

C. The Agent Security Facility

Network management in untrusted Internet environments imposes a thorough security framework, which should also be flexible enough to accommodate the range of MESIS operators with different levels of authorized operations, from network administrators to simple users. While one administrator may use MESIS for installing and configuring a network node, one user may develop and deploy a customized protocol to tailor the behavior and to optimize the performance in specific application scenarios.

This motivates the MESIS model of trust that defines who or what in the system is trusted, in what way, and to what extent [22]. MESIS has been developed for an untrusted Internet environment, where the communication network is considered insecure and any node may host the execution of possibly malicious entities. In addition, a MESIS agent is an active entity that

acts on behalf of a principal, i.e., the person/organization that has launched the agent execution and that is responsible for its operations. MESIS agents are authenticated by means of standard certificates, provided and administered through the integration with a public key infrastructure [38]; this integration permits agent authentication not only in the case of single-hop migration, but also when considering multiple-hop mobility. The actions that agents are authorized to perform depend on roles associated to agent principals. MESIS permits the dynamic definition and control of a range of roles, from administrators to users [23].

The MESIS security mechanisms support the model of trust and enforce security policies: authentication permits us to identify the role associated with MESIS agents; authorization recognizes whether an operation is permitted on a resource; integrity guarantees that agents and data have not been maliciously modified during reallocation; secrecy permits us to protect entities from any exposure to malicious intrusions.

In MESIS, security is provided with application level tools, taking advantage of available standard solutions and products (e.g., the IAIK cryptographic functionality and the Entrust Public Key Infrastructure [37], [38], [43]). If the debate concerning which level a system offers security is still open, the discussion concentrates on the issues of transparency, flexibility, and performance [13], [22]. Independently of the abstraction level adopted, it is important to consider security as a property to be integrated at any system layer. Only this pervasive approach followed by MESIS design can achieve the full level of security, higher than the minimal one obtained by systems that add an *a posteriori* security strategy.

The security infrastructure for mobile agents extends the traditional sandbox solution used to protect network nodes from the execution of untrusted code, because the sandbox approach limits too much the expressive power [43]. With regard to implementation, MESIS agents use X.509 certificates for authentication, which ascertain the role of the agent principal before authorizing any interaction with resources. We are currently working on the integration of MESIS with a commercial Public Key Infrastructure (PKI), provided by Entrust [38], to automatically distribute keys, to manage certificates, and to perform all related administrative tasks. The integrity check can employ either MD5 or SHA1. Secrecy is granted when needed by encrypting/decrypting communications with DES and SSL [44].

CORBA and MASIF standards recognize the security requirement by imposing tools and mechanisms to enforce security when interacting with external components. In accord with this guideline, MESIS addresses the security threats introduced by interoperating with CORBA. On one hand, sending/receiving CORBA requests/replies requires channel encryption to ensure privacy on exchanged messages. On the other hand, the possibility for MESIS agents to act as CORBA servers and for MESIS localities to host agents from other MA platforms calls for mechanisms for client/agent authentication, auditing, and access controlling. MESIS provides security solutions compliant with both CORBA security services and MASIF security specifications [30], [19]. We have also worked on providing MESIS compliance with the Secure Inter-ORB Protocol [30] to enable secure interactions between entities resident on different ORB's, provided that they adopt the same security technology.

Finally, MESIS gives users the possibility to choose the best tradeoff between security needs and required performance, according to the intended usage: agents in trusted environments (e.g., a private Intranet of a department) could directly access resources after the authorization check, while agents moving in untrusted environments (e.g., the Internet) generally have to pass all security steps for secrecy, integrity, authentication, and authorization.

V. Resource and Service Management in MESIS

MESIS provides a wide range of management tools: from the monitoring of the state of the distributed system to the possibility to control and coordinate replicated resources, from the dynamic installation and configuration of new network resources to the optimization of access to replicated information by considering both current traffic level and query locality. MESIS has demonstrated its suitability in implementing monitoring and controlling tools [12]. In addition, another goal of MESIS is to manage complex network services, even obtained by tailoring and composing existing ones, and to dynamically introduce new services in the existing infrastructure without suspending operations.

In the area of mobile computing, for example, we have tested MESIS usability for the design, implementation, and management of a Personal Communication Support service that answers the requirements of the Virtual Home Environment concept described by UMTS [45]. Another previously explored domain of MESIS applicability is, instead, the video on demand (VoD) application area reported in the following. The MESIS VoD service is based on a set of mobile lower-level services, implemented by agents that are distributed over the paths between the source and the targets of the video stream. MESIS VoD permits users to require a QoS level for any multimedia stream, and allows them to manage and adjust the requested quality during service provision, to respond to dynamic modifications of network resource availability.

The recent QoS research activity is exploring two different directions. On one hand, the definition and standardization of new protocols has been investigated to ensure the reservation of the needed amount of network resources [46], [47]. However, the process of acceptance and deployment of new standards for network-layer protocols is long and difficult, mainly due to the large base of nonprogrammable and already-installed network equipment. In this field, mobile agents have shown their suitability for implementing tunneling techniques to integrate network resources that are not compliant with the reservation standards [48]. On the other hand, some work has recently shown the opportunity of an application-layer approach to QoS, especially in the areas of mobile communications and multimedia distribution [49], [50]. Application-layer solutions propose service infrastructures that try to respect the specified QoS requirements without any guarantee of satisfaction, but with no need to modify the underlying best-effort network layer. The idea is to monitor the available QoS and to notify service components of quality modifications in order to adapt to the network

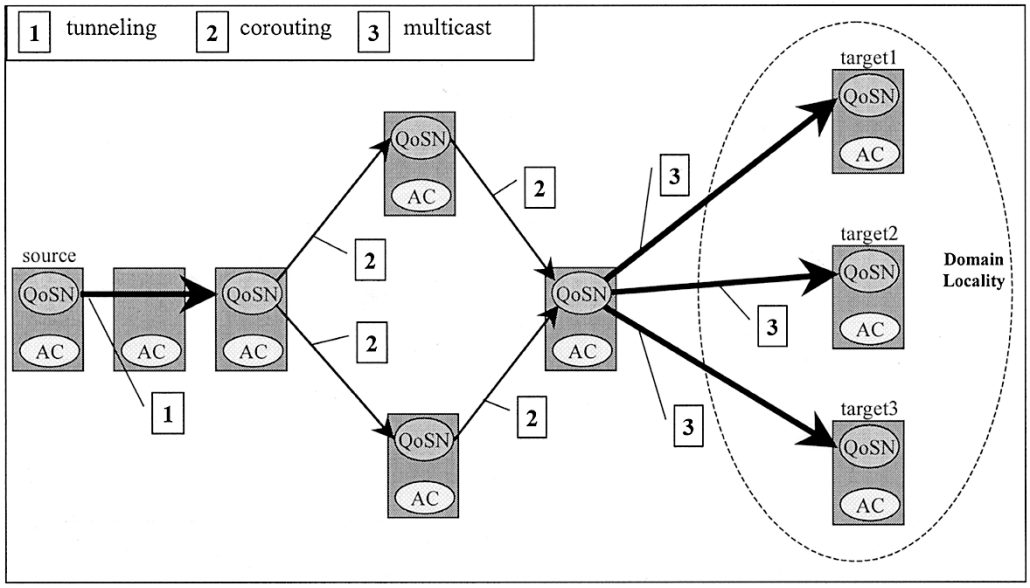

Fig. 5. Tunneling, co-routing, and multicast in the MESIS VoD service.

traffic. The MESIS VoD service adopts the application-layer approach, but we are extending its implementation to integrate network-layer technologies, such as ATM, that provide direct control of QoS parameters, with a solution that is similar to [51].

The VoD service is realized by coordinating two different types of MESIS management agents: the QoS negotiators (QoSN's) that define and grant a specific level of quality for the service, and the admission controllers (AC's) that manage the resources to be engaged by local intermediate nodes (see Fig. 5).

AC's 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 a local table of ⟨receiving-host, bandwidth, delay, loss⟩ tuples [52]. Any tuple represents the statistics of VoD 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 VoD sessions. AC's are in charge of answering to reservation requests from QoSN's.

The VoD service requires the coordination of a set of QoSN agents located at the source, at the target, and at some intermediate nodes. QoSN's maintain session state: they record user preferences and flow specifications for a video stream. QoSN's evaluate the feasibility of meeting these requirements against the local AC database and exploit the MESIS DPE communication facility to perform the negotiation phase for the definition of the achievable QoS. After the negotiation phase, during multimedia streaming, any QoSN is in charge of receiving packets from the previous QoSN and of forwarding them to the next QoSN. When multiple video streams interest the same network node, one QoSN can handle all of them.

Let us first consider the case of a video stream addressed to one target only. The path between the source and the target is automatically determined at runtime, by tracing the route via one dummy packet sent from source to target (it can be also predetermined by the VoD source according to some previously collected routing information). QoSN's move to the chosen hosts on the path and interrogate the AC database: if available resources are not enough for the desired QoS, QoSN's can coordinate and reduce their requests by scaling the stream (at the moment, by dropping frames in motion JPEG streams or by reducing resolution in MPEG-2 ones [52]). Only if these diminished reservation requests cannot be satisfied is the VoD service denied.

After a successful negotiation phase, the (possibly scaled) multimedia stream starts to flow. During the video distribution, a link can fail or its quality can deteriorate, thus making it impossible for a particular QoSN to maintain the negotiated quality. In that case, the interested QoSN can enhance the throughput of its link via stream striping on non-co-routed paths [53]. In this case, it sends back a message to temporarily stop the stream, and sends forward a message to suspend updates in AC tables on the path. Then, it sends its clones to handle new nonco-routed paths and starts the negotiation phase with the clones. When negotiation is completed, the QoSN sends back a message that restarts the stream: apart from a delay in receiving the stream, the VoD target goes on transparently.

In the case of multicast distribution of the same video stream (for N targets), the generated network traffic can be limited by exploiting location awareness of agents. While in traditional VoD systems the source generates N packet streams—one for each target—our QoSN's can ascertain whether there are several targets within the same domain locality, and can split packets only when it is necessary, in general only at the last hop. This is commanded by the QoSN at the gateway of the last domain. In a simple usage scenario with a set of homogeneous receivers, our multicast infrastructure provides a traffic optimization similar to the one achieved with multicast support at the network layer, but without the need for compliant hardware, e.g., for IP multicast routers. In a more complex scenario, our infrastructure can take advantage of its application-layer approach to perform

service-specific optimizations, such as video layering in the case of client-driven QoS adaptation for heterogeneous multicast receivers [54].

The MESIS VoD service requires the presence of the above described infrastructure composed by distributed and coordinated agents. Therefore, before the multimedia stream can start to flow, users have to wait for the completion of a setup phase in which the service provider determines the path between the source and the target, and distributes all AC's and QoSN's needed on participating nodes. 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, or to dynamically organize co-routed paths, with a further distribution of agents on new nodes. 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 in a normal usage scenario. In this phase, one lightweight agent (about 1 kb-sized) is sent from the source to the target to identify the path for the multimedia stream. This agent reports back to the source the information about how many AC's (about 6 kb-sized) and QoSN's (about 4 kb-sized) have to be instantiated and to be sent in parallel to interested nodes. We have considered the worst case when none of the intermediate nodes has neither the AC nor the QoSN agent. In more realistic scenarios, hosts may have already the AC agent running for purposes of remote monitoring and diagnosis.

In addition, the VoD service should be typically carried out in untrusted environments, where cooperating agents have to pass integrity and authentication checks before being allowed to operate to local resources. The MESIS architecture permits users to choose which subset of functionality are used by specific services. In that way, services can obtain the most suitable tradeoff between performance and security, depending on the level of trust of the environment and on the criticality of the application domain. In particular, the VoD service provided in a trusted environment may omit authentication and integrity checks, with considerable time saving on the setup overhead.

We report the setup time for the MESIS VoD service in a nondedicated network consisting of the interconnection of several LAN's. In particular, the results apply to a case where the video source is 8 hop far from the target, i.e., the multimedia flow has to pass through 7 intermediate nontunneled nodes to reach its target. Any intermediate node hosts the default place of the domain it belongs to, and any domain abstraction models a distinct real LAN in our university organization. The LAN's are composed by heterogeneous hosts (PentiumII PC's with Windows NT 4.0 and Sun SPARCstation with Solaris 2.5) and are based on different communication technologies, mainly Ethernet and Fast-Ethernet.

In the case of untrusted environments, the multimedia source has to calculate a 1024-bytes blocksize MD5 hash of all sent agents and to sign them with its 1024-bits RSA private key. Any intermediate node must perform the security checks to verify the signature and the hash. The average setup time measured in this scenario is 10 907 ms and shows the feasibility of the approach in the case of complex interconnection of nondedicated local networks.

In the case of trusted environments, MESIS gives the possibility of providing the VoD service with no security checks, with a considerably reduced average setup time (7634 ms that is 30% less of the first case). We are experimenting other significant time reductions via the utilization of the HotJava just-in-time compilation techniques, and via the recent introduction in our organization of fast communication technologies based on FDDI and ATM.

VI. Conclusion

Several new technologies propose efficient solutions for network, systems, and service management, each one facing a particular class of management problems. The management of open and global networks requires the integration of different solutions within one framework. In addition, new proposals should be capable of providing rapid application development of management tools and fast deployment of new network services. While the heterogeneity of existing components and legacy systems forces us to focus on the interoperability requirement, global untrusted environments require us to consider security as a basic property.

The MESIS framework proposes an integrated solution for the management of both resources and services in open and global environments, such as the Internet. MESIS permits the coexistence of different management paradigms through the DPE facilities made available by several DPE implementations. We have already developed some management tools and network services in MESIS, and our current work is directed to check the openness of the environment by verifying interoperability performance with different CORBA-based management systems, and to provide an interoperable and secure personal communications support for mobile computing applications.

References

[1] J. P. Thompson, "Web-based enterprise management architecture," *IEEE Commun. Mag.*, vol. 36, Mar. 1998.
[2] N. Anerousis, "An architecture for building scalable web-based management services," in *J. Networks Syst. Manage.*, Mar. 1999, vol. 7.
[3] Object Management Group. (1998, Feb.) CORBA/IIOP Rev 2.2. [Online]. Available: http://www.omg.org/library/c2indx.html
[4] P. Haggerty and K. Seetharaman, "The benefits of CORBA-based network management," *Commun. ACM*, vol. 41, no. 10, Oct. 1998.
[5] S. Mazumdar and K. Swanson, "Web based management—CORBA/SNMP gateway approach," presented at the 7th IFIP/IEEE Int. Workshop Distributed Syst.: Operations Manage., L'Aquila, Italy, Oct. 1996.
[6] UH Communications ApS. The UHC CORBA/CMIP Gateway product. [Online]. Available: http://www.uhc.dk/
[7] G. Goldszmidt and Y. Yemini, "Distributed management by delegation," presented at the 15th Int. Conf. Distributed Computing Syst., 1995.
[8] A. Fuggetta, G. P. Picco, and G. Vigna, "Understanding code mobility," *IEEE Trans. Software Eng.*, vol. 24, May 1998.
[9] A. Karmouch, Ed., "Special section on mobile agents," in *IEEE Commun. Mag.*, July 1998, vol. 36.
[10] M. Breugst, L. Hagen, and T. Magedanz, "Impacts of mobile agent technology on mobile communications system evolution," *IEEE Personal Commun.*, vol. 5, Aug. 1998.
[11] A. Bieszczad, B. Pagurek, and T. White, "Mobile agents for network management," *IEEE Commun. Surveys*, 1998.
[12] P. Bellavista, A. Corradi, and C. Stefanelli, "An open secure mobile agent framework for systems management," *J. Network Syst. Manage.*, vol. 7, no. 3, Sept. 1999.

[13] T. M. Chen and A. W. Jackson, Eds., "Special issue on active and programmable networks," in *IEEE Network Mag.*, May/June 1998, vol. 12.
[14] T. Magedanz and R. Popescu-Zeletin, Eds., *Intelligent Networks—Basic Technology Standards and Evolution*. London: Thomson Computer Press, June 1996.
[15] Y. Inoue, D. Guha, and H. Berndt, "The TINA consortium," *IEEE Commun. Mag.*, vol. 36, Sept. 1998.
[16] R. H. Glitho and S. Hayes, Eds., "Special issue on telecommunications management network," in *IEEE Commun. Mag.*, Mar. 1995, vol. 33.
[17] R. H. Glitho, "Contrasting OSI systems management to SNMP and TMN," *J. Network Syst. Manage.*, vol. 6, no. 2, June 1998.
[18] P. Bellavista, A. Corradi, and C. Stefanelli, "A secure and open mobile agent programming environment," presented at the Int. Symp. on Autonomous Decentralized Systems, Tokyo, Japan, Mar. 1999.
[19] D. Milojicic *et al.*, "MASIF: The OMG mobile agent system interoperability facility," in *Proc. 2nd Int. Workshop Mobile Agents*: Springer-Verlag, Sept. 1998, vol. 1477, Lecture Notes in Computer Science.
[20] D. L. Tennenhouse *et al.*, "A survey of active network research," *IEEE Commun. Mag.*, vol. 35, Jan. 1997.
[21] K. Psounis, "Active networks: Applications security safety and architectures," *IEEE Commun. Surveys*, 1999.
[22] R. Oppliger, "Security at the Internet layer," *IEEE Computer Mag.*, vol. 31, Sept. 1998.
[23] E. C. Lupu and M. Sloman, "Towards a role-based framework for distributed systems management," *J. Network Syst. Manage.*, vol. 5, no. 1, Mar. 1997.
[24] K. Rothermel and F. Hohl, Eds., *Proc. 2nd Int. Workshop Mobile Agents*: Springer-Verlag, Sept. 1998, vol. 1477, Lecture Notes in Computer Science.
[25] D. Gavalas *et al.*, "An infrastructure for distributed and dynamic network management based on mobile agent technology," presented at the IEEE Int. Conf. Commun., Vancouver, June 1999.
[26] "Special issue on agent technologies within intelligent networks and mobile communication systems," *Computer Networks*, vol. 31, no. 19, 1999.
[27] IKV++ GmbH—Grasshopper. [Online]. Available: http://www.ikv.de/products/grasshopper/.
[28] ObjectSpace—Voyager. [Online]. Available: http://www.objectspace.com/products/vgrOverview.htm
[29] P. Felber, R. Guerraoui, and A. Schiper, "The implementation of a CORBA group communication service," *Theory Practice Object Systems*, vol. 4, no. 2, 1998.
[30] Object Management Group. (1998, Dec.) CORBA Services. OMG Doc. formal/98-12-09 [Online]. Available: http://www.omg.org/corba/sectrans.html
[31] S. Staamann *et al.*, "Security in the telecommunications information networking architecture—The CrySTINA approach," in *Proc. TINA '97—Global Convergence Telecommun. Distributed Object Computing*, 1998.
[32] ISO/IEC, 10165-1, "Information technology—Open system interconnection—Structure of management information: Management Information Model,", CCITT Recommendation X.720, 1992.
[33] J. P. Redlich, M. Suzuki, and S. Weinstein, "Distributed object technology for networking," *IEEE Commun. Mag.*, vol. 36, Oct. 1998.
[34] P. Albitz and C. Liu, *DNS and BIND*, 3rd ed. Sebastopol, CA: O'Reilly & Associates, Sept. 1998.
[35] T. Howes and M. Smith, *LDAP: Programming Directory—Enabled Applications with Lightweight Directory Access Protocol*. New York: Macmillan Technical, Jan. 1997.
[36] S. C. Horstmann and G. Cornell, *Core Java 1.2: Fundamentals*. Englewood Cliffs, NJ: Prentice-Hall, 1998, vol. 1.
[37] Institute for Applied Information Processing and Communications, IAIK JCE. [Online]. Available: http://jcewww.iaik.at/jce/jce.htm
[38] Entrust Technologies, IAIK JCE. [Online]. Available: http://www.entrust.com/
[39] J. Park, S. Ha, and J. W. Hong, "Design and implementation of TMN SMK system using CORBA ORB," *J. Network Syst. Manage.*, vol. 6, no. 2, June 1998.
[40] Object Management Group, IAIK JCE. (1997, Apr.) Intelligent Networking using CORBA. [Online]. Available: http://www.omg.org/docs/telecom/97-04-02.doc
[41] D. Lange and M. Oshima, *Programming and Deploying Mobile Agents with Java Aglets*. Reading, MA: Addison-Wesley, 1998.
[42] Inprise—Visibroker, IAIK JCE.. [Online]. Available: http://www.borland com/visibroker/
[43] L. Gong *et al.*, "Going beyond the sandbox," presented at the USENIX Symp. Internet Technologies Syst., Monterey, CA, Dec. 1997.
[44] Institute for Applied Information Processing and Communications—IAIK iSaSiLk, IAIK JCE.. [Online]. Available: http://jcewww.iaik.tu-graz.ac.at/iSaSiLk/isasilk.htm
[45] UMTS 22.70, IAIK JCE, *Universal Mobile Telecommunication System (UMTS) Service Aspects: Virtual Home Environment v. 2.0.0*, Mar. 1998.
[46] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource ReSerVation Protocol," *IEEE Network*, vol. 7, Sept. 1993.
[47] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," *Computer Commun.*, vol. 19, no. 1, Jan. 1996.
[48] H. de Meer *et al.*, "Tunnel agents for enhanced Internet QoS," *IEEE Concurrency Mag.*, vol. 6, Apr. 1998.
[49] D. Chalmers and M. Sloman, "A survey of quality of service in mobile computing environments," *IEEE Commun. Surveys*, 1999.
[50] M. T. Kone and T. Nakajima, "An architecture for a QoS-based mobile agent system," presented at the 5th IEEE Int. Conf. Real-Time Computing Syst. Appl., 1998.
[51] A. Kassler, H. Christein, and P. Schulthess, "A generic API for quality of service networking based on Java," presented at the IEEE Int. Conf. Commun., Vancouver, June 1999.
[52] S. J. Gibbs and D. C. Tsichritzis, *Multimedia Programming*. Reading, MA: Addison-Wesley, 1994.
[53] C. B. S. Traw and J. M. Smith, "Striping within the network subsystem," *IEEE Network Mag.*, vol. 9, July/Aug. 1995.
[54] X. Li, M. H. Ammar, and S. Paul, "Video multicast over the Internet," *IEEE Network Mag.*, vol. 13, Mar. 1999.

Paolo Bellavista (S'97) received the Laurea degree in electronic engineering from the University of Bologna, Italy, in 1997. He is currently pursuing the Ph.D. degree in computer science engineering at the Department of Electronics Computer Science and Systems at the same university.

His research interests include distributed computing, distributed objects, mobile agents, network and systems management, adaptive multimedia systems, and distance learning.

Mr. Bellavista is a student member of ACM and AICA (Italian Association for Computing).

Antonio Corradi (S'78–M'78) received the Laurea degree in electronic engineering from the University of Bologna, Italy, in 1979 and the M.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1981.

He is currently an Associate Professor of computer science at the University of Bologna. His scientific interests include distributed systems, object and agent systems, network management, and distributed and parallel architectures.

Dr. Corradi is a member of ACM and AICA.

Cesare Stefanelli (M'98) received the Laurea degree in electronic engineering from the University of Bologna, Italy, in 1992 and the Ph.D. degree in computer science in 1996.

He is currently an Associate Professor at the University of Ferrara. His research interests include distributed and mobile computing, mobile code, network and systems management, network security.

Dr. Stefanelli is a member of USENIX and AICA.