

# QoS-aware Accounting in Mobile Computing Scenarios

Paolo Bellavista, Antonio Corradi, Silvia Vecchi

*Dipartimento di Elettronica Informatica Sistemistica - University of Bologna*

*Viale Risorgimento, 2 – 40136 Bologna – ITALY*

*Ph.: +39-051-2093001; Fax: +39-051-2093073*

*{pbellavista, acorradi, svecchi}@deis.unibo.it*

## Abstract

*The enlarging market of portable devices and wireless networks stimulates the provisioning of mobility-enabled Internet services with differentiated levels of Quality of Service (QoS). The finally supplied QoS level can greatly differ from the negotiated one because of the very variable availability of network resources. This significantly impacts on the pricing policies and, consequently, on the definition and realization of fair accounting strategies. The paper claims the need of middleware solutions able to evolve dynamically depending on user/terminal mobility to monitor, control and register the finally supplied QoS level directly within the network localities where users/terminals move to. The paper presents the design and implementation of the Active middleware for Quality-aware Accounting of Mobile services (AQuAM) that is capable of accounting final users for the QoS level finally supplied during service provisioning, in contexts of user/terminal mobility.*

## 1. Introduction

Recent advances in mobile telecommunications and portable device miniaturization propose new service. Recent advances in wireless connectivity and portable devices propose new service scenarios where users, access terminals and even service components can geographically move during provisioning. This forces to face several technical challenges at different levels of abstraction, from the integration of heterogeneous network technologies to user/terminal location tracking, from device-dependent service adaptation to location-aware service provisioning. In particular, service provisioning to mobile devices with wireless connectivity has to consider the wide heterogeneity of their hardware/software

characteristics and the highly dynamic variations of resource availability in the different localities that host the roaming terminals [1].

For instance, while approaching a city by train, a tourist with a laptop connected to the large bandwidth network offered to first class seats is likely to receive a high-quality multimedia stream about the historical buildings of the city. Already available Web servers can provide multimedia tourist the information for that service; however, these data should be dynamically filtered and adapted to fit both the client location and the bandwidth/visualization capabilities of the access device. On the contrary, while visiting the city downtown, the same tourist can be interested in receiving only fixed images and textual information (maps, small-size pictures, historical notes, ...) on her wireless personal digital assistant, also to reduce the time/costs of connectivity. Or, she would like to receive only textual information to her WAP-enabled phone, which cannot visualize either multimedia streams or fixed images. In any case, mobile scenarios stress the necessity of providing very differentiated Quality of Service (QoS) levels, determined dynamically depending on terminal characteristics, user preferences, and local resource availability.

The above scenario motivates the re-thinking of service management solutions in general, and, in particular, of accounting mechanisms and tools. Traditional management is tailored to fixed infrastructures of network elements and to QoS-insensitive packet communication; it is based on effective solutions and architectures to deal with geographic distribution and heterogeneity of resources and service components. IETF and OSI have proposed management models based on Client/Server (C/S) interaction and on several variations of it [2, 3]. It is possible to organize hierarchies of C/S components to achieve decentralization and scalability, but the interaction of management entities is usually statically determined for clients and servers in exchanging remote information. These solutions do not fit well global systems in rapid evolution where the unpre-

dictable entering/exiting of heterogeneous wireless devices is usual.

Accounting solutions in mobile scenarios should be capable both of handling properly the dynamic modifications of the client location, by identifying users independently of their position and used terminals, and of taking into account the possible change in the offered QoS level during service provisioning. At provision time, the final QoS may vary not only because of the best-effort model of the Internet communication, but also due to the dynamic change of user/terminal position and to the resulting modification in the resource availability. In fact, client mobility strongly impacts on the involved network resources, on the routing paths followed from the server to the client, and on the resource availability in the client locality. The QoS level negotiation at service setup is not sufficient to fairly account final users. Users are likely to pay for their final QoS level during service provisioning; it would be definitely necessary to measure the actually received QoS at the client side and to exploit the monitored QoS parameters to register, bill and audit the final users accordingly.

We propose a trade-off between the previously negotiated QoS and the actually received QoS: the finally supplied QoS. We use the term finally supplied QoS to indicate the level of quality offered at the end of the service provisioning chain, in the wired node providing service accessibility to the mobile user or to the mobile device. We claim the need for novel distributed infrastructures capable of following the user roaming, in case of both user and terminal mobility [1], in order to meter the finally supplied QoS level. These accounting functions are orthogonal to service provisioning and the infrastructure should support them transparently with regards to the application-level development. Only this separation of concerns can limit the complexity of the design and implementation of mobility/QoS-enabled services. In addition, the hardware/software resource constraints of several categories of wireless portable devices, from personal digital assistants to programmable cell phones, impose the realization of a dynamic and extensible accounting infrastructure over the fixed Internet hosts that act as points of attachment for the wireless terminals. This infrastructure should be capable of evolving during service provisioning depending on client mobility, by dynamically migrating middleware components close to the access terminals for monitoring, controlling and registering the received QoS levels in any network locality willing to open its resources to mobile accessibility.

Along these guidelines, we have designed and implemented the **A**ctive middleware for **Q**uality-aware **A**ccounting of **M**obile services (AQuAM). AQuAM disseminates new active components on the fixed network, when and where needed, to complement the typical limitations of mobile terminals. It extensively exploits

the Mobile Agent (MA) technology to install dynamically new monitoring components, to maximize locality in the access to monitoring data and in obtaining concise accounting indicators, and to achieve the complete decentralization of accounting, pricing and billing functions, without requiring continuous connectivity with remote and centralized home accounting managers. AQuAM is built on top of the Secure and Open Mobile Agent (SOMA)<sup>\*</sup> platform, exploits the SOMA facilities for the portable monitoring of distributed heterogeneous systems, and is organized in terms of accounting MAs that automatically follow the client roaming. AQuAM integrates with our previous research in network/systems/service management and mobile computing middlewares [4, 5]. First deployment experiences and experimental evaluations point out how the joint adoption of portable QoS monitoring mechanisms and MA-based middleware components make AQuAM a highly dynamic, effective and portable environment to obtain innovative accounting solutions for the emerging mobility-enabled Internet scenario.

## 2. Middleware Solutions for QoS-dependent Accounting

Service provisioning with differentiated quality for mobile accessibility forces to tackle several issues, from the negotiation of the suitable QoS level to the mechanisms for delivering the service to mobile users/terminals by preserving the agreed upon quality, from the definition of proper pricing/charging schemes to the implementation of accounting infrastructures enforcing the desired pricing/charging strategies.

The accounting management area addresses all the issues of metering, storing and processing information about resource consumption. Metering deals with measuring and collecting resource usage information. Storing maintains metering data on stable storage support, either in the same locality where information has been collected or in a remote administration domain with the additional related issues about transport and security. While metering and storing involve similar activities in all management applications, the processing phases may greatly vary because they deeply depend on the accounting goals, e.g., access control, auditing, capacity planning, and billing. This paper specifically focuses on accounting for billing and pricing, and, in particular, on QoS-enabled accounting in mobility scenarios.

In this context, the emerging guideline is to be able to bill customers for the negotiated upon QoS levels. That makes billing depend on the reserved amount of resources before service provisioning. However, there are many factors (currently available resources, used ac-

---

<sup>\*</sup> The SOMA code and additional information about the SOMA platform are available at: <http://lia.deis.unibo.it/Research/SOMA>

cess terminal and position, ...) that can greatly modify the QoS level ultimately provided to the paying customer. A fair strategy for providers is to make clients aware that they ultimately pay not for the negotiated QoS but for the effective level of QoS they obtained. This goal strategy can be achieved only by tracking the resource usage of all mobile users/terminals, and to account for the effective usage of them. For instance, a specified QoS level is dynamically offered only if there is a provisioning chain that permits to connect a potential receiver to the service provider with enough available resources, also in the intermediate traversed nodes, to support that QoS level.

Traditional management solutions, from the OSI Common Management Information Protocol (CMIP) [6] to the IETF Simple Network Management Protocol (SNMP) [2], face also accounting aspects, and, even if they are not specifically tailored for accounting management, have been widely exploited in several accounting systems. Accounting in the open Internet requires also some forms of authentication and authorization: IETF and IRTF have established working groups on Authentication Authorization Accounting (AAA) activities to deploy architectures and protocols targeted to provide AAA services for the Internet. IETF AAA has proposed two different solutions, Remote Authentication Dial In User Service (RADIUS) and Diameter [7, 8] [9]. RADIUS mainly focuses on how service components and authentication servers should exchange authentication, authorization and configuration information; it has been successively extended to consider accounting functions but does not address any mobility-related issue. On the contrary, Diameter deals with roaming and mobile IP networks, by maintaining the network-layer approach typical of the RADIUS solution. The IRTF AAA Arch solution specifically addresses the issues related to the dynamicity of requirements of different providers and focuses on the provision of a policy-based support infrastructure [10, 11, 12].

The above systems represent general-purpose management solutions and their diffusion preceded the interest in QoS and wireless. These areas ask for specific management functions that motivate both QoS-aware accounting and mobility-enabled resource consumption metering. Novel design solutions and implementations from scratch are needed and this has motivated several related research projects in the last months.

Market Managed Multiservice Internet (M3I) has the goal of designing a new system to enable Internet resource management on the basis of differentiated charging for multiple levels of service [13]. The Charging and Accounting System integrates the Java technology with SNMP and aims at differently charging applications that require different QoS levels, at dynamically modifying tariffs and promptly distributing them to involved users. The Next Generation Internet (NGI) accounting working group is focusing specifically on dif-

ferentiated charging strategies, capable of modifying even during Internet service provisioning: in particular, the NGI architecture for reverse charging enables an ISP to account final users for the received traffic, even if the users are currently connected to another ISP [14, 15, 16]. The Mobility and Differentiated Services in a future IP Network (MobyDick) defines and implements a mobility-enabled end-to-end QoS architecture based on IPv6 [17, 18]. With regard to accounting, MobyDick proposes an AAA Charging infrastructure that interfaces the DiffServ architecture with a dedicated Application-Specific Module (ASM); ASM permits to control service access and to account roaming terminals/users by adopting solution patterns similar to mobile IP.

The above research proposals deal with charging and accounting customers depending on the ultimately provided QoS level and most of them support some forms of user/terminal mobility. However, they tend to propose an approach at a low level of abstraction which does not move up to the application level. On the contrary, we claim the relevance of adopting accounting middlewares capable of covering also the application level to enable the application-specific processing of accounting data and to facilitate the integration with high-level functions, available infrastructure components and legacy systems.

### 3. AQuAM: a QoS-dependent Accounting Middleware for Mobile Scenarios

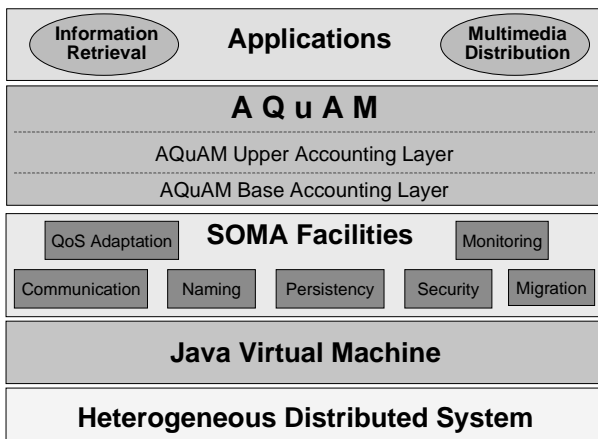
The **A**ctive middleware for **Q**uality-aware **A**ccounting of **M**obile services (AQuAM) is a layered support infrastructure capable of accounting final users for the QoS level ultimately supplied during service provisioning, in context of user/terminal mobility. AQuAM works at the middleware level to answer both the need of having visibility of service-oriented data typically not available at the network level, e.g., customer preference and security profiles, and the need of propagating low-level accounting information up to the application level to enable corresponding service reactions, e.g., modifying QoS level depending on the currently experienced frame rate in a VoD service.

To better explain the AQuAM accounting solution, it is useful to distinguish preliminary between negotiated and finally supplied QoS level. The *negotiated QoS level* is the level that customer and provider agree up on; it is defined at service startup and can be different from the provided QoS when dealing with a best-effort network infrastructure. On the contrary, we define *finally supplied QoS level* the level provided at the end of the provisioning chain. In case of user mobility the provisioning chain terminates at the fixed terminal the user exploits to access the service; in case of terminal mobility the provisioning chain ends at the point of attachment to the fixed network that provides wireless con-

nectivity to the mobile device. The received QoS level at the user side coincides with the finally supplied one in case of user mobility; in case of terminal mobility, instead, the two QoS levels may differ because of the further provisioning step between the fixed node and the mobile device.

AQuAM accounts users for the finally supplied QoS level. This requires to measure the relevant QoS parameters at the end of the service provisioning chain, by taking into account the possible user/terminal movements, i.e., by tracking the client migrations and changing dynamically the QoS measurement points. The AQuAM middleware achieves this objective by exploiting the Mobile Agent (MA) technology. MAs are intrinsically suitable for dynamic changes of allocation: MAs can migrate by following mobile users/devices in their de/attachment to the fixed network, and can deploy accounting functionality where and when needed during service provisioning. Moreover, the property of maintaining the execution state when detaching from a previous location and of restoring it so to execute from there in their new location can significantly help in tracing the service usage history.

AQuAM operates on top of Secure and Open Mobile Agent (SOMA). SOMA is a general-purpose, layered mobility middleware that provides a rich set of facilities (see Figure 1) to assist in the design, implementation and deployment of MA-based services for global, open and untrusted environments. SOMA offers locality abstractions to describe any kind of interconnected system, from simple Intranet LANs to the Internet. Any node hosts at least one *place* for agent execution; several places can be grouped into *domain* abstractions that correspond to network localities. In each domain, a *default place* is in charge of inter-domain routing functionality and integration with legacy components. The architecture and functionality of the SOMA middleware are extensively described in [1] [19].



**Figure 1. The AquAM layered architecture on top of the SOMA middleware**

The AQuAM infrastructure consists of two main components: the Basic Accounting Layer and the Upper Accounting Layer. The former furnishes the accounting mechanisms and solutions for resource metering and accounting data storage: it is the basis of all accounting functions and is dynamically deployed in any managed host over the fixed network infrastructure. The latter adopts the MA technology to distribute the proper accounting strategy where and when needed at service provisioning time: MA-based accounting managers can determine the QoS level from monitoring data depending also on the specific service class, can collect accounting information off-line, and can allow service usage on the basis of user profiles.

### 3.1. Basic Accounting Layer

The Basic Accounting Layer includes two main services: the Monitoring Service and the Storage Service. The Monitoring Service provides information about resource consumption both at the system level and at the application one [5]. At the system level, it gets information on the processes acting on local resources and on their usage of the communication infrastructure. At the application level, it collects information about all service components accessed from within the Java execution environment. The monitoring data are used to ascertain the finally supplied QoS level. For instance, in a multimedia streaming service, a crucial QoS parameter is the frame rate. In case of multimedia flows with constant size frames, it is possible to compute the frame rate from the number of received packets per second by working with monitoring data at the system level. In case of variable size frames, it is possible to operate at the application level, for instance by triggering a monitoring event any time the Java method for frame reception and processing is invoked. These metering operations are performed on the wired hosts providing service accessibility to mobile users/terminals and possibly adapting service provisioning to access devices with limited capabilities.

The Storage Service provides a non volatile support for the accounting data processed by the MA-based accounting managers, described in the following. This accounting information describes service usage (QoS level and duration) of customers at specific hosts, either when users access directly the service at that host, i.e., in case of user mobility, or when the specified host has supported the network connectivity of the users' portable devices, i.e. in case of terminal mobility. The persistent storing of accounting information enables off-line accounting processing and overcomes data loss in case of network failure/partitioning, as briefly sketched in Section 4.

### 3.2. Upper Accounting Layer

The Upper Accounting Layer exploits the base one and determines how, where and when to perform accounting management. It is implemented in terms of two types of MA-based accounting managers: *Home Agents* (HAs) and *Tracking Agents* (TAs). When a user requests a service from a fixed/mobile terminal, AQuAM associates the user with two care-of entities: one HA and one TA. HA resides in the SOMA place where the user starts requesting a service. HA is in charge of coordinating accounting activities related to its user depending on the enforced policies for service provisioning and user charging. TA follows the user in its migrations during service provisioning: TA is in charge of gathering information from the local monitoring, of processing the accounting data in accordance with the user profile, and of managing them by coordinating with the corresponding HA.

Policies for service provisioning and user charging are negotiated between users and providers, and are maintained at the SOMA directory service, together with user profiles [1]. User profiles include personal data about identity, service preferences, e.g., required QoS level for specific applications and tariff plan. Users and service providers can register user profiles and service accounting policies to AQuAM by exploiting a registration/updates interface. Any default place maintains a directory server (Accounting Profile Directory) where profiles and policies are partially replicated and partitioned, thus transparently decentralizing this information in the AQuAM infrastructure.

The management of accounting data is obtained via the interaction of the accounting managers. TA sends the results of accounting data processing to its HA immediately after any locality change of its client (and consequently before any of its migration). In case of network partitioning, TA can work autonomously: it registers the processed accounting results at the storage service and, only at the end of the service session, recovers the results and communicates them to HA. HA traces the history of the service usage performed by its assigned client by collecting the information received from TA.

The main advantages of this approach are the distribution of the computing of accounting data elaboration, the reduction of the amount of transmitted data, and the complete decentralization in the exchange of accounting management messages. TA processes locally the accounting information in any visited node, and sends to HA only elaborated and summarized data, by significantly reducing the generated network traffic. In addition, the accounting management of any user is performed at the first point of attachment of the service provision session, by avoiding the problems of a unique central node in charge of managing all users.

## 4. AQuAM Implementation Insights

Two crucial activities in the AQuAM infrastructure are the mechanisms for the local monitoring of system/application QoS parameters and the strategy for the collection of distributed accounting data.

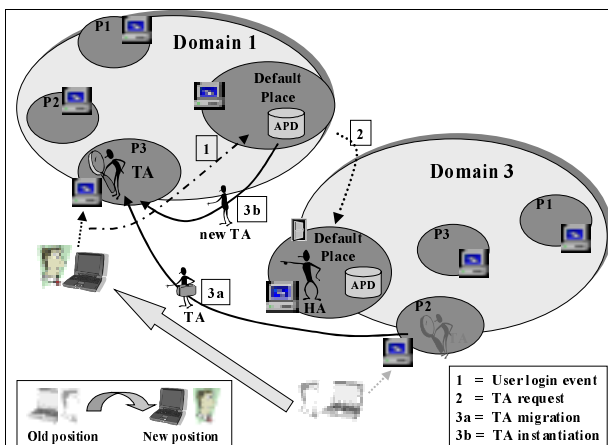
As briefly sketched above, the AQuAM Monitoring Service provides information about the resource consumption of a single host both at the system level, e.g., allocated memory, network packets and file operations, and at the application level, e.g., invoked methods by Java threads. For any process, the Monitoring Service reports the process identifier and name, the CPU usage (time and percentage, of the process and of its composing threads) and the allocated physical/virtual memory. TAs can ask the Monitoring Service for CPU and memory utilization during service provisioning anytime the finally supplied QoS level can be estimated from those monitoring data, e.g., in a service where the node at the end of the provisioning chain performs complex filtering and transcoding operations on the data flow before delivering it to the client [20]. More usually, TAs ask for network traffic information including the total number of sent/received UDP packets, of sent/received TCP segments, of TCP connections, and of TCP/UDP packets received with errors. At the application level, instead, for any active Java thread on the local Java Virtual Machine, the Monitoring Service provides lifetime, number of loaded classes and used monitors, number and size of allocated objects, number of TCP/UDP and file read/write operations, and also number of invocations of service-specific application-level Java methods. The monitoring data are collected by exploiting extensions of the Java technology such as the Java Virtual Machine Profiler Interface and the Java Native Interface. The overhead introduced by monitoring activities can be maintained lower than 3% in a large set of usual deployment scenarios [5]. In addition, let us note that it does not affect the portable client devices with limited computational resources, but only more powerful nodes, such as multimedia adaptation servers and gateway hosts providing accessibility to wireless clients. For additional details about monitor implementation and performance see [5].

HAs and TAs collaborate to perform the collection of distributed accounting data on the basis of the local monitoring information provided by the Monitoring Service. Let us present an example of mobility-enabled deployment scenario to give a better idea of how TAs and HAs interoperate (see Figure 2). When one user U, from one either fixed or mobile terminal, enters a new domain, her entrance generates an event notification (1), which triggers a sequence of accounting actions. Firstly, the system ascertains whether U has already one open service session within AQuAM, i.e., it controls whether U started to access the service in a previously visited

domain (domain 3 in the figure). To achieve this goal, the middleware retrieves the involved HA from the AQuAM Accounting Profile Directory (APD), if not already known, and notifies HA of the new U's location (2).

Then, in the case in the figure, HA triggers the TA migration from domain 3 to domain 1 (3a). In the worst case of network partitioning, when it is not possible to reach the U's HA, the middleware sends a multicast search message to all known neighbors to ultimately discover a running TA associated with U. All receiving nodes recursively ask other adjacent nodes by exploring the SOMA inter-domain connections. In the case no answers are received within a specified interval, a brand new TA is automatically instantiated (3b). In that way, AQuAM permits to overcome network fault situations, for the sake of availability, by allowing several contemporary TA instances associated with the same user. When the connectivity is reestablished, HA is in charge of collecting all the accounting information by the different TA instances, independently of their positions.

TA collects and elaborates monitoring data to determine the finally supplied QoS level at the host over the fixed network that provides connectivity to user U. Only when U either disconnects or changes her place of attachment, TA sends the processed accounting information to HA. In case TA cannot communicate the data to HA, e.g., in network partitioning conditions, TA stores locally the accounting data, to be collected asynchronously later when the connectivity will be reestablished.



**Figure 2. AQuAM accounting components at work in an example of deployment scenario.**

## 5. Conclusions

Mobility forces to re-think traditional support solutions for distributed applications and motivates the need for innovative functions to be provided by novel middlewares. In particular, Internet service provisioning to mobile users/terminals stresses the relevance of tailoring

differentiated QoS levels and demands for middleware accounting functions capable of both dealing with client mobility and considering the actually client-received QoS. AQuAM shows how an highly dynamic and extensible active middleware infrastructure can support service accounting in mobility scenarios, without requiring the intervention of developers over the service implementation. In addition, the AQuAM project points out that the adoption of the MA technology and of portable monitoring is a viable and effective way to implement a wide variety of accounting, pricing and billing strategies, determined only at service provision time and enabled even in case of network partitioning.

The first encouraging deployment experiences and experimental evaluations of our accounting middleware are stimulating further work to extend the current prototype and to apply it in different application domains. In particular, we are currently extending AQuAM to support the accounting of groups of users taking part to the same multicast tree of service provisioning in the multimedia distribution domain, by considering fair cost-sharing strategies for final users with different QoS levels received.

## Acknowledgements

Work supported by the Italian Ministero della Ricerca Scientifica e Tecnologica in the framework of the Project "MUSIQUE: Infrastructure for QoS in Web Multimedia Services with Heterogeneous Access" and by the University of Bologna Funds for Selected Research Topics "An Integrated Infrastructure to Support Secure Services".

## References

- [1] P. Bellavista, A. Corradi, C. Stefanelli, *Mobile Agent Middleware to Support Mobile Computing*, IEEE Computer, Vol. 34, No. 3, Mar. 2001.
- [2] J. D. Case, et al., *Simple Network Management Protocol (RFC 1157)*, DDN Network Information Center, SRI International, May 1990.
- [3] G. Dickson, A. Loyd, *Open Systems Interconnection*, Trevor Housley (Ed.), Prentice-Hall, 1992.
- [4] P. Bellavista, A. Corradi, C. Stefanelli, *An Integrated Management Environment for Network Resources and Services*, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 5, May 2000.
- [5] P. Bellavista, A. Corradi, C. Stefanelli, *How to Monitor and Control Resource Usage in Mobile Agent Systems*, 3<sup>rd</sup> IEEE Symp. Distributed Objects and Applications (DOA'01), Sept. 2001.
- [6] International Organization for Standardization, *Management Information Protocol Specification – Part 2: Common Management Information Protocol*, ISO IS9596, Jan. 1990.

- [7] C. Rigney, A. Rubens, W. Simpson, S. Willens, *Remote Authentication Dial In User Service (RADIUS)*, IETF RFC 2138, Jan. 1997.
- [8] C. Rigney, *RADIUS Accounting*, IETF RFC 2139, Jan. 1997.
- [9] P. R. Calhoun, P. Pan, H. Akhtar, *Diameter Framework Document*, <http://search.ietf.org/internet-drafts/draft-ietf-aaa-diameter-framework-01.txt>
- [10] The Authentication, Authorization, Accounting Architecture Research Group, <http://www.aaaarch.org/>
- [11] J. Salowey, G. Sliepen, A. Taal, D. Spence, *Policies in AAA*, draft-irtf-aaaarch-aaa-pol-01.txt, March 2001
- [12] A. Taal, G. Sliepen, C.T.A.M. de Laat, *A Grammar for Policies in a Generic AAA Environment*, draft-ietf-aaaarch-generic-policy-02.txt, March 2002.
- [13] The Market Managed Multiservice Internet Project, <http://www.m3i.org/index.html>
- [14] The Internet Next Generation Project, <http://ing.ctit.utwente.nl>
- [15] A. Pras, B. Van Benijum, R. Sprenkels, R. Parhonyi, *Internet Accounting*, IEEE Communication Magazine, May 2001.
- [16] R.A.M. Sprenkels, R. Parhonyi, A. Pras, B.J. van Benijum, B.L. De Goede, *An Architecture for Reverse Charging in the Internet*, Proceedings of the IEEE Workshop on IP-oriented Operations and Management (IPOM00), Cracow, Poland, Sept. 2000.
- [17] The Mobility and Differentiated Services in a future IP Network, <http://www-int.berkom.de/~mobydick>
- [18] H. Einsiedler, R. L. Aguiar, J. Jaehnert, K. Jonas, M. Liebsch, R. Schmitz, P. Pacyna, J. Gozdecki, Z. Papir, J. I. Moreno, I. Soto, *The Moby Dick Project: A Mobile Heterogeneous ALL-IP Architecture*, ATAMS'01, Cracow, Poland.
- [19] P. Bellavista, A. Corradi, *A Mobile Agent Infrastructure for Terminal, User and Resource Mobility*, IEEE/IFIP Network Operations and Management Symposium (NOMS 2000), Apr. 2000.
- [20] F. Baschieri, P. Bellavista, A. Corradi, *Mobile Agent for QoS Tailoring, Control and Adaptation over the Internet: the ubiQoS Video on Demand Service*, 2<sup>nd</sup> IEEE International Symposium on Applications and the Internet (SAINT'02), Japan, Jan. 2002.