Mobile-to-Mobile Multimedia Service Provisioning in the IMS Using REST-Based Mobile Services

Muzzamil Aziz Chaudhary and Matthias Jarke

***Abstract*—With the merging of IP and TelCo networks, Mobile-to-Mobile (M2M) social networks are to be unfolded as an ultimate telecom business in the near future. By M2M, both service provider and service client are aimed to be mobile based. Hence considering mobile devices to be resource constrained and mobile networks as low-bandwidth networks, provisioning high-profile multimedia services is not somewhat straight forward. The Quality of Service (QoS) has always been a great concern in this scenario. Additionally in M2M service networks, the direct IP accessibility of mobile devices is a bottleneck in seamless communication. The cellular operators typically block every incoming http access to subscribers devices inside and outside the network due to huge traffic load and security reasons.**

The paper addresses the issues faced in M2M service networks and presents a solution based on the IP Multimedia Subsystem (IMS). Similar to SIP Application Servers (AS) in the IMS network, the concept of Mobile Application Servers (Mob-AS) is introduced in order to host and execute multimedia services from a mobile device. A comprehensive design of Mob-AS is presented based on the existing REST interfaced synchronous and asynchronous mobile server framework.

***Index Terms*—M2M social networks, mobile application server, quality of service, synchronous, asynchronous.**

I. Introduction

M2M service networks are the mobile networks where a subscriber device can offer and consume services to and from other subscriber devices on the network. The paper deals with the operators cellular networks in particular which typically are limited bandwidth networks compared to high speed WLANs. Therefore from the network perspective, hosting high-profile multimedia applications over such networks is not straight forward. QoS has always been an issue in such scenario and network over-provisioning is not an economical solution. Thus, the operators have to employ more sophisticated control and policy enforcement mechanism in order to ensure the quality of individual services running over their network. More precisely, they require a precised way to do service differentiation and apply QoS policies accordingly.

Moreover, providing direct IP access to mobile devices is also a bottleneck in provisioning M2M service networks. The operators typically employ firewalls to prevent http access to subscribers' devices over the internet. This fact can be observed from a recently conducted research explaining how M2M multimedia streaming went through over two big operator networks, O2 and T-Mobile. Table I depicts the summary of live network transactions [1]. It is observed that in case both client and server devices are connected on private IP addresses, M2M streaming is not possible until unless an intermediate access gateway is involved. Furthermore, the direct streaming went successful in case the server is on public IP address with the exception that operator restricts the session explicitly.

TABLE I: M2M Streaming Results over Live Networks

Client Network	Client IP	Gateway	Server Network	Server IP	Result
O2 (HSPA)	Private	NA	T-Mobile (HSPA)	Private	Stream Failed
T-Mobile (HSPA)	Private	NA	O2 (HSPA)	Private	Stream Failed
O2 (HSPA)	Private	Relay	T-Mobile (HSPA)	Private	Successful
T-Mobile (HSPA)	Private	Relay	O2 (HSPA)	Private	Successful
O2 (HSPA)	Private	Relay/ NA	T-Mobile (HSPA)	Public	Successful
T-Mobile	Private	NA	T-Mobile (HSPA)	Public	Successful
T-Mobile	Private	Relay	T-Mobile (HSPA)	Public	Failed (Operator restricts)

Nokia in its research of Mobile Web Server also pointed out the addressability issue of mobile devices over operator networks and proposed a solution based on intermediate access gateway [2].

The IP Multimedia Subsystem (IMS) is now being seen as a de-facto for service layer architecture in the mobile LTE systems [3]. The initiative has been taken by 3GPP in order to bring the mobile operators into the services business by providing them a full command and control of next generation interactive and interoperable multimedia services. The proposal facilitates the mobile operators to host multimedia services on their network and ensure guaranteed QoS for their provisioning. The paper deals with the aforementioned issues of M2M service networks and presents a solution by introducing a concept of Mobile Application Server (MobAS) in the IMS network and presenting a mobile server platform compliant to IMS standards defined by 3GPP. Section II details how an IMS compliant framework facilitates seamless M2M service delivery over an operator network. Section III discusses the QoS mechanism employed by 3GPP evolved packet system (EPS). Section IV introduces the concept of Mobile Application Server (MobAS) and explains the mechanism of M2M service invocation. Section V gives a comprehensive design of REST interfaced M2M framework whereas Section VI presents conclusion.

Manuscript received January 15, 2013; revised March 28, 2013.

The authors are with the Chair of Computer Science 5 - Information Systems RWTH Aachen University, Ahornstr. 55, D-52056 Aachen, Germany (e-mail: {muzzamil, jarke}@dbis.rwth-aachen.de).

II. IMS – Application Servers

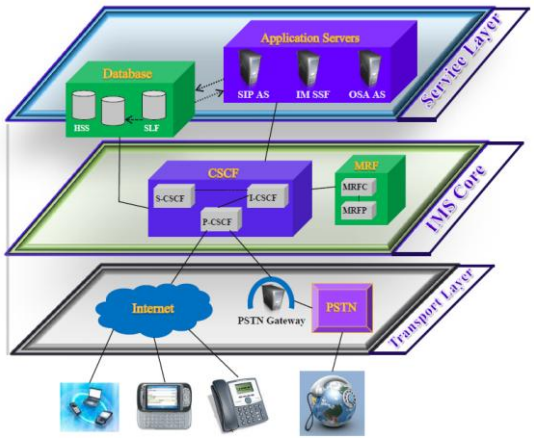

Fig. 1. 3GPP IP multimedia subsystem.

A high-level view of 3GPP-IMS ecosystem is shown in the Fig. 1. For simplicity, the whole architecture is divided into three different layers [3]: Transport layer, IMS Core and Service Layer. It is important to note that 3GPP specifies the IMS architecture as a collection of functions not nodes, leaving the choice to vendors to define single or multiple nodes to implement these functions. Figure shows that the so-called IMS Core is accessible using any user device connected to the internet via WLAN, ADSL or packet data network such as GPRS using a radio link. A user device capable of communicating to and from IMS Core is typically referred to as an IMS terminal. The transport layer facilitates an IMS terminal to make and receive calls to and from PSTN network or any other circuit-switched network using PSTN/CS gateway as shown in figure.

The IMS Core contains one or more SIP proxies, collectively known as CSCFs (Call/Session Control Functions), MRFs (Media Resource Functions) and user databases, called HSSs (Home Subscriber Servers) and Subscriber Location Functions (SLFs). The CSCF is the most essential part of the IMS Core. It controls all the signaling information to and from IMS terminals and Application Servers (ASs). A CSCF can be one of the three different types: P-CSCF (Proxy-CSCF), I-CSCF (Interrogating-CSCF), S-CSCF (Serving-CSCF).

The Service layer contains the entities responsible for hosting and executing multimedia services in the IMS network, typically referred as ASs (Application Servers). An AS can be found in three different types [4]: SIP AS, open service architecture (OSA) AS and CAMEL AS. As obvious by names, each server type is dedicated for a specialized type of services. Although some basic requirements are identified, 3GPP does not specify the underlying architecture of these servers. However, some software designs have been presented by research communities based on the state-of-the-art APIs and technologies targeted mainly for enterprise solutions such as [5].

III. Quality of Service Control in the 3GPP Evloved Packet System

Multimedia services due to their delay-sensitive nature require special treatment over bandwidth-constrained cellular networks as compared to rich-bandwidth fixed IP networks. Over-provisioning typically is an uneconomical solution in this regard due to relatively high cost for transmission capacity in cellular access networks (including radio spectrum and backhaul from the base stations) [6]. Hence, in order to provide special treatment to multimedia sessions, an efficient mechanism is required by cellular operators to differentiate among subscribers and service data. In this manner, an operator will be able to control each individual data session and perform appropriate actions to attain required performance. The similar idea has been standardized in 3GPP Release 8 for policy and charging control (PCC) in the LTE evolved packet system (EPS) [6].

The EPS utilizes its PCC architecture to control QoS in the operator data network. Its QoS concept is shown in Fig. 2. It is important to note that the IMS is an essential part of this architecture which is playing a critical role of session establishment (SE). Here, the IMS Core has the key responsibility to identify the session type, prepare QoS treatment and signal this information to the network gateway.

The Fig. 2 illustrates a use case where a subscriber sets up an IMS voice call. The subscriber's SIP request comes to the CSCF in order to establish the session. Thus CSCF first checks whether the subscriber is authorized to this call and then transfer the request to the callee. Upon receiving an Ok message by callee's CSCF, the CSCF sends the session information to the policy controller (PCRF) using an AAR message. The PCRF in turn selects PCC rules from the policy database and prepares QoS information to be applied to this particular session. This information is then transferred to the network gateway in an RAR message. Finally, the gateway establishes the bearer according to the information received.

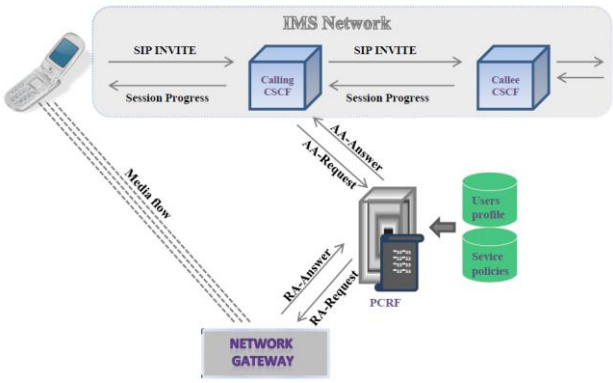

Fig. 2. QoS control in the LTE-evolved packet system

IV. Mobile-to-Mobile Service Provisioning

This section describes that how the IMS architecture helps the subscribers' devices to make one-to-one contact over the operator cellular network. In this paper, we propose that M2M services are to be invoked in a client server fashion where one (client) of the two communicating parties makes request and the other party (server) serves the request and invokes the requested service. For every short or long process to be performed by server, there exists a service categorized as synchronous (short-term) or asynchronous(long-term) respectively.

A two way communication is shown in Fig. 3 where both devices are capable of becoming service client and server at the same time by using mobile server platform (MSP). The MSP is comprised of the IMS mobile application client (MobAC) and the IMS mobile application server (MobAS). Similar to SIP ASs in the IMS network, the MobAS is a mobile based SIP application server capable of hosting and executing multimedia services. Section V describes the design of MSP in detail. The figure depicts a chat messaging session between two mobile applications, A1 and A2. The application A1 requests for an image file from A2. The application server AS2 receives a SIP request message from the client AC1, invokes its messaging service and sends a 200 Ok response back to AC1. The application A2 reads the message and picks an image file to be sent to A1. Accordingly, the application client AC2 gets an acceptance from application A1 by sending a SIP request message and receiving response from the server AS1 in the same fashion as above. Until this step (no. 10), the messaging service discussed is a fine example of synchronous mobile service which implements a simple short-lived process i.e., receiving the client request and sending response back immediately without keeping the client in a blocked state for long.

Having the initial negotiation done, the application client AC1 sends a SIP INVITE request to the server AS2 in order to invoke actual media sharing. The AS2 sends a 200 OK response immediately and invokes a multimedia messaging service later on after having an ACK message from AC1. At this point, a direct connection is setup between the service and the client AC1 and packets start forwarding from service to AC1. Finally, the service closes the connection with a SIP Bye request after the final packet is sent. The multimedia messaging service in this scenario is an example of asynchronous mobile service because the process of sharing multimedia files is somewhat a lengthy process depending on the media file size.

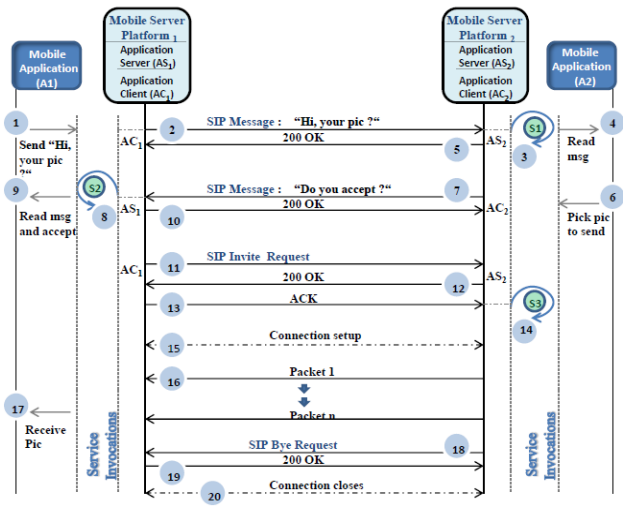

Fig. 3. M2M chat application scenario

Now, the question is where and how the IMS plays role in this scenario and helps the devices to keep direct contact over the operator network. This discussion could be divided into two main phases: Session establishment (SE) phase and media delivery (MD) phase.

Session establishment phase: In the chat messaging example shown in figure, the SE phase is comprised of steps 1-14. Here, simple message sharing is done in order to establish a media session without the actual media being transferred between two applications. These are the messages sent over the IMS network and go through the SIP proxies, the CSCFs. However, every single terminal is required to upload its current location (network address) before a session startup by registering to the IMS core network. Therefore, the communication between two mobile devices is provisioned via SIP proxies in the SE phase requiring no direct connection among them.

Media delivery phase: The steps 14-17 denote the MD phase in figure which is solely responsible for actual media transfer among the communicating parties. The media is transferred through the operator's network gateway which involves no IMS SIP proxy in between sending and receiving devices. Therefore, a special mechanism is required here to make the devices directly contact over the operator's IP network. So, there are two possibilities to solve this problem. First, like every application server in the IMS network the operator should provide a direct access to the user devices acting as MobAS either by introducing special policies or by assigning public IPs to them. Second, operators should utilize full strength of IMS by shifting their system to IP V6 as described in [7].

V. Mobile Server Platform / M2M Framework

This section explains the design and architecture of Mobile Server Platform (MSP) as shown in Fig. 4.

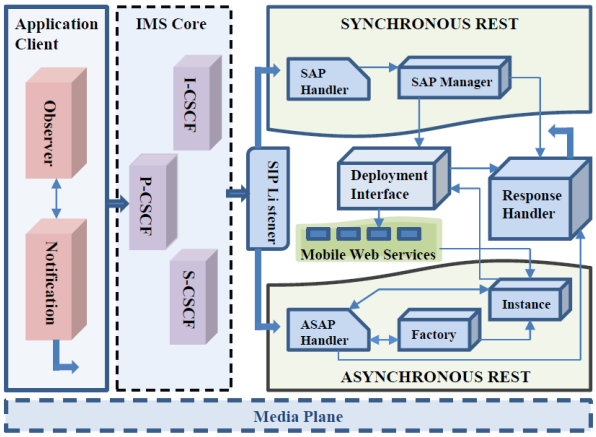

Fig. 4. Mobile server platform.

A. Application Client

Observer: Observer is an integral component of MobAC which is responsible to create and send synchronous and asynchronous service calls. The synchronous service calls are exposed based on the request-response interaction policy of WSDL standard [8]. In a request-response process each client request is followed by a response. Therefore, the Observer keeps itself in a blocked state after a synchronous service call until the response is back from the requested service invoked by the MobAS. On the contrary, the mechanism of asynchronous service calls is derived by incorporating the request-response and solicit-response WSDL operations. Similar to request-response operation, solicit-response is a

process where every request is followed by a response, but the roles of service provider and service consumer are reversed. In contrast to synchronous calls, the Observer does not wait for a complete service response after an asynchronous service call, rather an acknowledgement is sent by the MobAS to the Observer that the request is received. However, during this request-response operation the Observer provides an End Point Reference (EPR) to the MobAS where the service outcome shall be notified. After this operation, the service starts running independently in a separate thread sending response back to the provided EPR in the form of short notifications with partial results time to time or a single complete response at the end of successful task completion. This process of result notification in asynchronous service calls is analogous to solicit-response operation. Typically, asynchronous interaction is utilized to realize multimedia messaging and streaming services, where the initial request-response operation is used to create session with an additional acknowledgement message and solicit-response is used for actual media delivery by the service.

B. SIP Requests Binding with REST URI

In accordance with REST design principle, the mobile server platform exposes services interaction and invocation based on RESTful URIs. Thus, a new header field is introduced in the SIP request format for this purpose, named as rest-uri. Conforming to the standard, a RESTful URI is a simple HTTP URL where the URL is not only used to specify the location of the resource but also conveys the actions that must be performed at the serving node. For example, the GET method in the REST URL demands some information from the system without changing its behavior whereas the POST method demands a change in the system with the payload information in the HTTP Packet.

Two types of SIP Requests based on synchronous and asynchronous system are discussed below. For more details on REST URI format for synchronous and asynchronous services, see [9].

Synchronous SIP requests: The SIP request methods used to create synchronous service calls are termed as Synchronous SIP requests. For example, SIP MESSAGE and SIP Bye requests are used to invoke s1 and s2 synchronous services in Fig. 3. Table II shows a synchronous SIP request where a SIP user Alice wants to send a text message to another SIP user Bob. It is noticeable that the request contains a 'rest-uri' header with Restful URI pointing to a synchronous service for Instant Messaging, IMSync.

TABLE II: SYNCHRONOUS SIP REQUEST

```
MESSAGE sip:greetings@bob-server.com SIP/2.0
Max-Forwards: 69
CSeq: 1 MESSAGE
resourcemethod:GET
Content-Length: 84
rest-uri: POST/IMsync
Contact: "alice"
<sip:alice@10.0.2.15:5070>
To: "bob" <sip:bob@ericsson.com>
From: "alice" <sip:alice@ericsson.com>;tag=12345
Call-ID: 79a6e89a0f8c55849222730b2a90769f@10.0.2.15
Via: SIP/2.0/udp 127.0.0.1:5082
Content-Type: text/xml
<?xml version='1.0' ?><RESTRq><Rq><demo>Hi
Bob, alice here!</demo></Rq></RESTRq>
```

Asynchronous SIP requests: The SIP request methods used to create asynchronous service calls are termed as Asynchronous SIP requests. For example, SIP INVITE request is used to invoke an asynchronous service s3 in Fig. 3. Table III shows an asynchronous SIP request where a SIP user Alice wants to send a multimedia file to another SIP user Bob. Here the 'rest-uri' header points to an asynchronous service for multimedia Instant Messaging, IMasync.

TABLE III: ASYNCHRONOUS SIP REQUEST

```
INVITE sip:greetings@bob-server.com SIP/2.0
Max-Forwards: 69
CSeq: 1 INVITE
resourcemethod:GET
Content-Length: 751
rest-uri: POST/Request-Response/Factory/
createInstanceRq/IMasync/add
Contact: "alice"
<sip:alice@10.0.2.15:5071> To: "bob"
<sip:bob@ericsson.com>
From: "alice" <sip:alice@ericsson.com>;tag=12345
Call-ID: 37c08f9666a687a3027ac22b98483158@10.0.2.15
Content-Type: application/sdp Via:
SIP/2.0/udp 127.0.0.1:5082
v=0 o=coco 1326930823911 1326930823911 IN
IP4 alice@ericsson.com
s=sdp offer (recvonly)
c=IN IP4 alice@ericsson.com t=0 0
m=message 9097 TCP/MSRP
a=recvonly:recvonly
a=accept-types:message/cpim a=accept-wrapped-types:*
a =path:msrp://alice@ericsson.com:9097/jshA7we;tcp
a=file-name:multimedia.jpg
a=file-transfer-id:aCQYuBRVoUPGVsFZkCK98vzcX2FXDIk
2 a=file-size: 38503
```

C. Application Server

As in Fig. 4, the SIP Listener separates the Restful-URI string from the service request and parses it to extract the target service information such as service name, request method and resource method. The parsed information is then utilized by SIP Listener to form a Request Object and identify the service type such as synchronous or asynchronous. It is worth mentioning here that although the service type can also be identified by simply knowing the SIP method name, such as, SIP MESSAGE corresponds to synchronous request and SIP INVITE corresponds to asynchronous, but this approach is avoided in order to keep compliance with basic REST architecture introduced in [9].

D. Synchronous Service Invocation

In case the synchronous service request identified by SIP Listener, the Request Object is passed to the Deployment Interface through SAP Manager which is responsible to look up and invocation of target service. The Deployment Interface maintains a list of all available services along with their corresponding objects as a key-value mapping. Therefore, it looks up the corresponding service object in its service inventory and uses it to invoke target service.

E. Asynchronous Service Invocation

An asynchronous service is initiated and executed as a thread, which is the primary difference between the synchronous and asynchronous services. The asynchronous service is made a thread because that enables it to be controlled and monitored and to be in variable states during the course of execution [9]. In case of asynchronous service request, before parsing and making Request Object the SIP Listener first checks whether the request message contains an

SDP offer, extracts the media information and verifies the availability of the target service. This is to note that the SDP offer is used in the SIP INVITE request in order to negotiate over streaming media parameters between two parties before the actual media transmission starts. The SDP works in conjunction with RTP and MSRP protocols for real-time streaming and multimedia instant messaging services. Considering the SDP offer is valid and accepted by the SIP Listener, the Request Object is passed to ASAP Handler which invokes Service Factory to create a new Service Instance for the target service. At this stage, a 200 OK message is sent to the AC that releases the AC from the blocked state. With this message the AS also provides two important IDs to the AS: a request Call-ID for reference to the upcoming final acknowledgement from the AC and the EPR for newly created Service Instance for future communication with the service.

Thus, after sending 200 OK message the control is given to the Deployment Interface finally along with the Request Object and Service Instance parameters for target service invocation. In contrast to synchronous service invocation, after inventory look up the Deployment Interface forms a service thread, save it in database against the Call-ID and wait for the acknowledgement of the last 200 Ok message from the AC. Consequently, when the acknowledgement message arrives at Request Handler, the Deployment Interface is notified along with the Call-ID which in return starts the corresponding service thread.

VI. Summary

Telecom operators today are more interested in business system which enables them to apply multiple charging policies over their network based on the user subscriptions and service profiles. The system must ensure QoS provisioning of high-profile multimedia services such as Push-To-Talk, video conferencing and multiparty gaming over the operator network. Keeping such demand by mobile users and network operators in view, 3GPP introduced the concept of IMS to guarantee QoS of multimedia applications over the operator network. By addressing the main issues faced in M2M service networks, the paper proposes a M2M service platform based on IMS ecosystem. The paper introduces the concept of mobile application server in the IMS network in order to host and execute multimedia services over a mobile device. The service invocation mechanism in M2M scenario is explained using the proposed M2M service platform whereas two different solutions are presented to counter the IP addressability issue for mobile devices.

References

[1] A. S. Chaudry, "Design and evaluation of mobile-to-mobile multimedia streaming using rest-based mobile services," Master Thesis, KTH, the Royal Institute of Technology, January 26, 2011.

[2] J. Wikman and Dosa. (May 2006). Providing HTTP access to web servers running on mobile phones, Nokia Research Centre Helsinki. [Online]. Available: http://research.nokia.com/tr/NRC-TR-2006-005.pdf

[3] The architecture and benefits of IMS. White Paper by Dialogic Corporation Canada. [Online]. Available: http://www.dialogic.com/~/

[4] H. Khalifi and J. Gregoir, "Ims application servers roles, requirements, and implementation technologies," *IEEE Internet Computing magazine*, vol. 12, no. 3, pp. 40-51, May-June, 2008.

[5] H. Khlifi, Dialexia, and J. Gregoire, "Ims for enterprises," *IEEE Communication Magazines*, vol. 45, no. 7, pp. 68-75, July 2007.

[6] H. Ekstrom, "Qos control in the 3 g pp evolved packet system," *IEEE Communication Magazines*, vol. 47, no. 2, pp. 76-83, February 2009.

[7] I. Ali, A. Kasati, K. Chowdhury, and K. Nishida, "Network-based mobility management in the evolved 3gpp core network," *IEEE Communication Magazines*, vol. 47, no. 2, pp. 58-66, February 2009.

[8] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana. (May 2005). Web services description language (wsdl) version 2.0 part 1: Core language. [Online]. Available: http://www.w3.org/TR/2005/WD-wsdl20- 20050510

[9] F. Aijaz, S. Z. Ali, M. A. Chaudhary, and B. Walke, "Enabling high performance mobile web services provisioning," in *Proc. the 2009 IEEE 70th Vehicular Technology Conference Fall*, Anchorage, Alaska-USA, pp. 6, December.

Muzzamil Aziz Chaudhary is a research assistant and PhD student at the chair of Computer Science 5 – Information Systems, RWTH Aachen. He received his master's degree in December 2008 from KTH, Stockholm Sweden. From November 2008 to December 2010, he worked as a research assistant at the Department of Communication Networks (ComNets), RWTH Aachen. He is currently working in the area of service layer architecture and technologies of mobile communication systems.

Matthias Jarke is a professor of Information Systems at RWTH Aachen University and Executive Director of the Fraunhofer FIT Institute for Applied Information Technology. He is founder director of the Bonn-Aachen International Graduate Centre for Information Technology (B-IT) which is supported by RWTH Aachen, the University of Bonn and the Fraunhofer-Gesellschaft. Jarke holds master degrees in business administration and computer science and a doctorate in business informatics, both from the University of Hamburg, Germany. Prior to joining Aachen, he held faculty positions at New York University's Stern School of Business and at the University of Passau. His research area is information systems support for cooperative activities in business, engineering, and culture. He has been coordinator of three European research projects in these areas. He is currently deputy coordinator of the German national excellence cluster in mobile communications in Aachen. Jarke has published about 25 books and over 250 refereed papers. Jarke was a Chief Editor of the journal Information Systems from 1993–2003, and has served as program chair of major international conferences such as VLDB, EDBT, CoopIS, and CAiSE. He is elected senior reviewer for software engineering for the German national science foundation DFG. From 2004–2007, he was president of the German Informatics society, GI.