User Personalization of Services in IMS

Mauro Fernandes Soares Sardinha

Dissertação para obtenção do Grau de Mestre em
Mestrado Integrado em Engenharia Electrotécnica e de
Computadores

Júri
Presidente: Prof. Dr. Nuno Cavaco Gomes Horta
Orientador: Prof. Dr. António Manuel Raminhos Cordeiro Grilo
Vogais: Prof. Dr. Luís Eduardo Teixeira Rodrigues

Outubro de 2009
Acknowledgments

First of all, I would also like to thank Prof. António Grilo, who guided the writing of this Thesis, for his kind suggestions and careful remarks. His support, patience and dedication certainly made this Thesis easier to write.

I would like to express my gratitude to Dr. Anett Schülke for allowing me to do this Thesis under her supervision at NEC Laboratories Europe, in Heidelberg, Germany. Working at NEC was an enriching experience I won’t soon forget. I also thank all the colleagues I had and all friends I made at NEC and Heidelberg without whom this work could not have been so successful.

I would like to thank all my friends and colleagues who have always been by my side through all these years. And last, but not least, a big thank to my Parents and brother, for the unconditional support, encouragement and love they gave me throughout all my academic career so far, particularly while away in Germany.
Abstract

The IP Multimedia Subsystem (IMS) is a functional network architecture currently seen as a promising solution for facilitating multimedia service creation and deployment, as well as supporting interoperability and network convergence. In this architecture a User can be reachable not only under different public identities but also on different devices. However, even if the User has only one device connected to the network he will probably like to personalize what he’s receiving as multimedia, and from whom he’s receiving it, among other options.

The Converged IP Multimedia (CPM) Enabler, under Open Mobile Alliance (OMA) standards, is to date the most significant step towards an optimal exploitation of the IMS capabilities. CPM is also the only driving source where advanced user preferences as the ones just described are being suggested for future services in the form of a User Preferences Profiles (UPP).

This Master Thesis investigates the motivation for a UPP together with the OMA CPM proposed protocols, profile technologies and related proposals; and possible alternatives. A design is presented and a supporting architecture implemented within an IMS infrastructure by combining the following protocols: Extensible Markup Language (XML), XML Document Management (XDM) and XML Configuration Access Protocol (XCAP). A working prototype of the UPP was also constructed with two of its preference parameters.

This Thesis thus constitutes a good first step in personalizing the user experience in IMS and opens up new avenues for further work in this topic.

Keywords

IP Multimedia Subsystem (IMS), User Personalization of Services, User Preferences Profile (UPP), XML Document Manager (XDM), XML Configuration Access Protocol (XCAP), Web Interface
Resumo

O IMS é uma arquitetura de rede funcional vista actualmente como uma promissora solução para facilitar a criação e implantação de serviços multimédia, tal como para suportar interoperabilidade e a convergência de redes. Nesta arquitectura um utilizador pode ser contactado não apenas sob diferentes identidades públicas mas também em diferentes aparelhos. No entanto mesmo que o utilizador tenha apenas um aparelho ligado à rede ele vai gostar de personalizar o que é que está a receber como serviços multimédia, e de quem, entre outras opções.

O CPM Enabler, especificado pela OMA, é até ao momento o passo mais significativo para uma exploração óptima das capacidades do IMS. O CPM é também a única força motora onde preferências do utilizador tão avançadas estão a ser sugeridas para serviços futuros.

Esta Tesse de Mestrado investiga a motivação para um UPP juntamente com os protocolos propostos pela OMA, tecnologias para perfis e propostas relacionadas; e possíveis alternativas. Um design é apresentado apresentado e uma arquitetura de suporte implementada dentro de uma infraestrutura IMS combinando os seguintes protocolos: XML, XDM e XCAP. Um protótipo funcional do UPP foi também construído com dois dos seus parâmetros de preferências.

Esta Tesse constitui portanto um bom primeiro passo para personalizar a experiência do utilizador no IMS e abre novos caminhos para trabalho futuro neste tema.

Palavras Chave

IP Multimedia Subsystem (IMS), Personalização de Serviços pelo Utilizador, User Preferences Profile (UPP), XML Document Manager (XDM), XML Configuration Access Protocol (XCAP), Interface Web
# Contents

1 Introduction and Objectives ........................................... 1
   1.1 Concept of Personalization ........................................ 3
   1.2 Contribution and Objectives ....................................... 4
   1.3 Structure of the Thesis ............................................ 5

2 State of the Art ......................................................... 7
   2.1 Service Delivery Platform (SDP) ................................... 8
   2.2 IP Multimedia Subsystem (IMS) .................................... 9
   2.3 Open Mobile Alliance Converged IP Messaging (OMA CPM) ........... 12
       2.3.1 OMA CPM User Preferences Profile (UPP) .................. 12
   2.4 Profiling Technologies .............................................. 14
       2.4.1 OMA CPM Shared User Profile (SUP) ......................... 15
       2.4.2 W3C Composite Capabilities/Preferences Profile (CC/PP) .... 18
       2.4.3 3GPP Generic User Profile (GUP) .......................... 19
   2.5 Implementation Alternatives for User Profiles .................... 20
       2.5.1 OMA XML Document Manager (XDM) ......................... 20
       2.5.2 XML Configuration Access Protocol (XCAP) ................ 23
       2.5.3 MySQL and PHP .............................................. 25

3 Technological Environment ............................................. 27
   3.1 OpenIMS .............................................................. 28
   3.2 Mobicents ............................................................ 29
   3.3 XDM Servers .......................................................... 30
       3.3.1 Mobicents XDM Server ........................................ 30
       3.3.2 OpenXCAP ..................................................... 31
   3.4 Java EE 5 .............................................................. 32
       3.4.1 Extensible Markup Language (XML) .......................... 32
       3.4.2 XML Schema ................................................... 33
       3.4.3 Java Server Pages (JSP) ...................................... 33
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>High level schema of UPP decision</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Simplified Architecture of the UPP Implementation</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Schematic of State of the Art</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Vertical and Horizontal Application Layer Architecture, source 8</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>IMS Architecture, adapted from source 12</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>CC/PP example code</td>
<td>18</td>
</tr>
<tr>
<td>2.5</td>
<td>GUP System Architecture, source 23</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Detailed View of XDM Architecture, source 26</td>
<td>21</td>
</tr>
<tr>
<td>2.7</td>
<td>XCAP root and Document Selector schema example</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>The Open Source IMS Core in the Open IMS Playground, source 31</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>Mobicents High Level Diagram, source 33</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Converging SOA and UOA, source 9</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>NEC IMS Application Layer, simplified from source 9</td>
<td>37</td>
</tr>
<tr>
<td>4.3</td>
<td>PCC Architecture Design, source 9</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>NEC SDP Prototype Deployment, simplified from source 9</td>
<td>39</td>
</tr>
<tr>
<td>5.1</td>
<td>Initial levels of depth of the XML Schema</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>XML Schema of the Shared User Sub-Profile’s</td>
<td>45</td>
</tr>
<tr>
<td>5.3</td>
<td>XML Schema of the UPP</td>
<td>46</td>
</tr>
<tr>
<td>5.4</td>
<td>UPP’s Supporting Architecture</td>
<td>50</td>
</tr>
<tr>
<td>5.5</td>
<td>UPP Interface Flowchart</td>
<td>52</td>
</tr>
<tr>
<td>6.1</td>
<td>Possible future version of UPP’s Supporting Architecture</td>
<td>60</td>
</tr>
</tbody>
</table>
List of Tables

5.1 Summary of evaluation on profile format technologies .................................. 42
B.1 Home UPP .................................................................................................. 70
B.2 Working UPP .............................................................................................. 71
B.3 Meeting UPP ............................................................................................... 71
B.4 Driving UPP ................................................................................................ 72
B.5 Sleeping UPP ............................................................................................... 72
B.6 Roaming UPP ............................................................................................... 73
B.7 Public Places UPP ......................................................................................... 73
B.8 Personal Mobile UPP ..................................................................................... 74
B.9 Personal Desktop UPP ................................................................................. 74
B.10 Personal Labtop UPP .................................................................................. 75
B.11 Work Mobile UPP ....................................................................................... 75
B.12 Work Desktop UPP ..................................................................................... 76
B.13 Work Labtop UPP ....................................................................................... 76
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>3GPP2</td>
<td>3rd Generation Partnership Project 2</td>
</tr>
<tr>
<td>AS</td>
<td>Application Server</td>
</tr>
<tr>
<td>CC/PP</td>
<td>Composite Capabilities/Preferences Profile</td>
</tr>
<tr>
<td>CPM</td>
<td>Converged IP Multimedia</td>
</tr>
<tr>
<td>CSCF</td>
<td>Call Session Control Functions</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise JavaBeans</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>GUP</td>
<td>Generic User Profile</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HSS</td>
<td>Home Subscriber Server</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messenger</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Subsystem</td>
</tr>
<tr>
<td>JSLEE</td>
<td>Java Service Logic Execution Environment</td>
</tr>
<tr>
<td>JSP</td>
<td>JavaServer Pages</td>
</tr>
<tr>
<td>NGN</td>
<td>Next Generation Networks</td>
</tr>
<tr>
<td>OMA</td>
<td>Open Mobile Alliance</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>PCC</td>
<td>Personalized Communication Control</td>
</tr>
<tr>
<td>PHP</td>
<td>PHP: Hypertext Preprocessor</td>
</tr>
<tr>
<td>PoC</td>
<td>Push to Talk over Cellular</td>
</tr>
<tr>
<td>RA</td>
<td>Resource Adapter</td>
</tr>
<tr>
<td>RAF</td>
<td>Repository Access Function</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>RMI</td>
<td>Remote Method Invocation</td>
</tr>
<tr>
<td>SBB</td>
<td>Service Building Blocks</td>
</tr>
<tr>
<td>SDP</td>
<td>Service Delivery Platform</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>SIP for Instant Messaging and Presence Leveraging Extensions</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SUP</td>
<td>Shared User Profile</td>
</tr>
<tr>
<td>TISPAN</td>
<td>Telecommunications and Internet converged Services and Protocols for Advanced Networking</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UOA</td>
<td>User Oriented Architecture</td>
</tr>
<tr>
<td>UPP</td>
<td>User Preferences Profiles</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Networks</td>
</tr>
<tr>
<td>XCAP</td>
<td>XML Configuration Access Protocol</td>
</tr>
</tbody>
</table>
List of Acronyms

**XDM**  XML Document Management

**XDMC**  XML Document Management Client

**XDMS**  XML Document Management Server

**XML**  Extensible Markup Language
1

Introduction and Objectives

Contents

1.1 Concept of Personalization ................................................. 3
1.2 Contribution and Objectives .............................................. 4
1.3 Structure of the Thesis .................................................... 5
1. Introduction and Objectives

Nowadays the massive proliferation of mobile devices and the innovation in mobile telecommunications is strongly promoting new value-added services. Although the Internet community is well ahead in handling these services, new architectures continue to be proposed to make these services available to the whole telecommunications world.

This Thesis discusses the aspects of communication services in an environment enabled with converged service delivery technologies for Next Generation Networks (NGN) based on IMS [1].

The NGN can be imagined as a packet-based network where the service/call control intelligence is physically and logically separated from the packet switching and transport elements, e.g., routers, switches, and gateways. This control intelligence is used to support all types of services ranging from basic voice telephony services to data, advanced broadband, and media services [2].

NGN based on IMS will bring many benefits to network operators. Operators that deploy it can potentially offer a wider range of applications and services to subscribers, independent of their location, and at reduced cost and complexity. IMS will allow operators to increase their revenues through the additional services that can be offered, while their operation expenses will decrease with their migration from legacy Time Division Multiplexing (TDM) networks, associated with telephone company voice services, to converged network architectures. With the increasing penetration of Wireless Local Area Networks (WLAN)s and emerging Wireless Metropolitan Area Networks (WiMax) as access network technologies, IMS scope is now extended for any IP access network, including fixed access networks like Passive Optical Network (PON) or Digital Subscriber Lines (DSL). The later aspect in particular also has driven NGN standardization bodies, such as the European Telecommunications Standards Institute (ETSI) and the International Telecommunications Union - Telecommunications Sector (ITU-T), to consider the IMS as an important service control platform for their NGN standards.

The IMS architecture allows the User to register with multiple devices and be contacted using one single public address. Communication sessions can be addressed to the User through his public address; however, if he has more than one device, then it must be decided to which one the session is routed. This decision should reflect the device capabilities (e.g., video capabilities), the context in which the User is accessing the IMS architecture (e.g., at home, at work, while driving, etc), and the User’s personal preferences for each device.

The work described in this Thesis was developed in collaboration with NEC Laboratories Europe on site in Heidelberg, Germany, for 8 months. Here a prototype of OMA’s CPM Enabler is being developed for R&D and proof of concept for various Service Delivery Platform (SDP) and IMS features. For convenience, this prototype will from now on be referred to as NEC SDP project.

The Figure 1.1 illustrates how the decision module in NEC’s SDP project named Personalized Communication Control (PCC), crosses the User’s Context, the Capabilities and UPP of each of the User’s devices with the communication parameters of incoming sessions, to filter out unfit
1.1 Concept of Personalization

devices. This filtering allows communication sessions to be routed to the device that best provides the service to the User.

![Diagram of UPP decision process]

This Thesis focuses on the last parameter set, UPP, as a new value-added service to NEC’s SDP project and as a contribution to the UPP standardization process at the OMA.

1.1 Concept of Personalization

At the MEX conference (27 and 28th May 2008, London), the mobile industry discussed a ten point Manifesto for enhancing the mobile experience among which, “Users as individuals: uniquely complex and contradictory”.[3]

The issue of personalization was summed up in the fact that customers can and will no longer allow to be defined by number or segments or demographics. Every User is uniquely complex and contradictory, thus creating an urge to design experiences which recognize customers as individuals. Research tools and analysis techniques must be developed to live and breathe the world as Users see it.

As an example let’s consider a business man that travels a lot. He is profiled by telecommunication companies as “senior executive, world traveler” in its market segmentation model. When
1. Introduction and Objectives

his contract comes up for renewal, the retention strategy is already planned out to offer him the newest Blackberry handset at no charge, a great deal on roaming and a free Bluetooth headset. It’s a market-leading anti-churn package and it’s sure to appeal to someone in his segment. However, since he travels a lot and is far away from home maybe what he actually wants is to see his young kids on a video call. This will lead him to buy another device from the “youth market” range. And because number portability came in, he has no particular loyalty to his current provider. The Blackberry has always made him feel too conventional anyway and he’ll be sure to enjoy his “youth” mobile. This will cause a major revenue loss to the current operator.

Humans are individually complex and never fit into a single segment. Devices should be adaptable for multiple profiles, adapting to context and diverse lifestyle.

It is in this context of need for user personalization of services that the UPP for IMS users and their devices is being proposed by OMA.

A UPP stores several parameters called preferences and is set to each user device. Profiles can be designed having in mind the device’s purpose (e.g., personal PDA profile) or user activity at the moment (e.g., driving profile), or any other criteria the User may desire. Preferences range from simple options like ringing loudness and roaming to more complex options, such as who can and cannot contact the User and to allow or not video and sound communication sessions, among other parameters.

1.2 Contribution and Objectives

This Thesis intends to make the following contributions to the solution of the problem that was stated in the previous section:

• Research and conceptual work for the implementation of a UPP integrated in an IMS architecture;

• Contribution (in the context of NEC) to the standardization process of a UPP for CPM at the OMA. To maximize this objective, the solution must stay as close as possible to the proposed technologies under standardization for user personalization of services;

• For this contribution the author of this Thesis has to plan and design a UPP and develop the necessary modules and connections for it to be integrated in the NEC SDP project. The whole process needs to be coordinated with the NEC team so a smooth integration can happen. The modules to be built by author of this Thesis are depicted inside the dashed line in Figure 1.2. The PCC and remaining modules are features already implemented in the NEC SDP project.

– An open source implementation of an XML Document Management Server (XDMS) must be deployed to work as a network-based storage server for the UPPs;
1.3 Structure of the Thesis

The remainder of this Thesis has the following structure:

- Section 2 gives an overview of the current state of the art technologies and standardization activities in the area of service delivery for converged services and personalization. Alternative implementations for user personalization of services are also presented.

- Section 3 gives an overall explanation of the main architectures used in the NEC’s SDP project. Available implementations of the technologies debated in Section 2 are also discussed.

- Section 4 elaborates on the motivations behind NEC’s SDP project, its main module (PCC) features and on its high level implementation.

- Section 5 reports about the designed UPP, implementation work done for it and details the Use Case created.

- Section 6 presents the main conclusions of this Thesis and lists possible future work.

An end-user interface named UPP Interface is necessary to fully interact with the XDMS and test the implementation; and

The UPP Manager module will link the pre-existing modules in the PCC with the new XDMS and thus enable a full integration of this Thesis work.

A working prototype is also intended for proof of concept. For this at least one preference must be implemented in the PCC, a Use Case designed and a demonstration video of it made.
1. Introduction and Objectives
2 State of the Art

Contents

2.1 Service Delivery Platform (SDP) .................................................. 8
2.2 IP Multimedia Subsystem (IMS) .................................................... 9
2.3 Open Mobile Alliance Converged IP Messaging (OMA CPM) ........... 12
2.4 Profiling Technologies ................................................................. 14
2.5 Implementation Alternatives for User Profiles ............................. 20
2. State of the Art

Figure 2.1 depicts the relationship between most of the architectures and technologies described in this Section. Note that only the concepts necessary to understand this thesis will be explained (i.e. not the technologies used elsewhere in the NEC SDP project).

The term SDP refers to a new system architecture designed to facilitate the creation and maintenance of a service, while the IMS answers these needs by supporting interoperability, network convergence and multimedia services. A UPP is a user profile detailing his preferences regarding access to the services, e.g. whether to receive video or sound communication sessions. The XDM is a standard by the OMA that specifies how user-specific service-related information (e.g. a UPP) is stored in logical repositories in the network, called XDMS in XML format, and how it is accessed and manipulated by the XCAP.

Besides these, other state of the art technologies and standardization activities in the area of service delivery for converged services and personalization important for the understanding of this thesis, as well as implementation alternatives are described in this Section.

2.1 Service Delivery Platform (SDP)

The term SDP refers to a system architecture that enables the efficient creation, deployment, execution, orchestration and management of one or more classes of services [4]. The SDP has emerged as a consequence of telecom network evolution. In the converging telecom world, a need to substitute numerous network-specific “stove-pipes” with a common, horizontal, service architecture seems right and natural and rose from several factors. Overall penetration values of market close to 100% limiting the number of subscribers, declining voice and data revenues and slowing growth in average revenue per User are a looming crises for all service providers. Besides fierce competition of other telecom companies, Internet/Application Service Providers and retailers are starting to bypass traditional telecom value chains and business models, thereby increasing the need for attractive services in order to maintain subscriber revenue. Because stealing customers from telecom competitors is only a patch to the problem, the current imperative is to change or be left behind.
Companies that can rapidly create and deliver new value-added services can maximize potential revenues, combat commoditization (when a product becomes indistinguishable from the competition and consumers buy based on price alone) and decrease customer churn (customers changing to similar products). To acquire a unique and diverse feeling, operators need to manage different types of services, such as operator-controlled, 3rd party provided and user-generated services. In parallel, industry analysts commonly agree that, in the future, Users will define their own set of preferred applications.

In light of this, new SDPs will need to be independent from access network technologies and will have to provide consistent device and network independent service experience to their Users. Existing services need to be enhanced and new ones deployed over converged networks and multiple service platforms. In that sense, the SDP will become a true Information and Communication Technologies (ICT) application.

To comply with the SDP requirements of reliability, scalability and performance, and to avoid operational difficulties resulting from the large number of components, the solution must integrate in one or a few platforms a significant number of components. In that context, a unique solution framework supporting an integration environment - where best-of-breed elements can be associated with solution components and specific developments - is the way to comply to the emerging market demand for SDP.

### 2.2 IP Multimedia Subsystem (IMS)

The IMS is a network functional architecture seen as a promising solution for facilitating multimedia service creation and deployment, as well as supporting interoperability and network convergence [5].

Using IMS as the basis for the NGN service layer in an all-IP network offers new opportunities to address the most interesting aspects of the SDP challenges:

- **Fixed Mobile Convergence (FMC)**

  Because IMS is the common standard for next generation fixed Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), cable (PacketCable) and mobile (3rd Generation Partnership Project (3GPP), 3rd Generation Partnership Project 2 (3GPP2)) networks, it can be deployed as a unique IMS core network for fixed and mobile access. Also, by using Session Initiation Protocol (SIP) it is possible to allow user-level convergence, because SIP allows the association of independent public user identities to a single User, and for these identities to be concurrently registered from multiple devices using different access technologies. The all-IP integrated network also fa-
2. State of the Art

Cilitates network management and has cost-related advantages, both from a CAPEX\(^2\) and an OPEX\(^3\) perspective.

- Multimedia Communications

SIP is a very generic rendezvous mechanism that allows services to use their own user-plane protocol(s), with SIP just setting their framework. A SIP session can be re-negotiated at any time thereby enabling media components to be added, removed or replaced at any time. With these attributes an IMS infrastructure will allow operators to deploy multimedia communication between IMS (and other SIP) endpoints.

- A new Service Platform Approach

The service approach (depicted in Figure 2.2) on SDP solutions is currently changing from a vertical approach with a stove-piped service model, with a voice-centric approach, to an horizontal one\[^7\]. The latter makes full usage of SIP and IMS capabilities with its generic session control layer. Consequently, it supports converged multimedia communication and content/data services across different IP domains.

- Service Oriented Architecture (SOA) versus User Oriented Architecture (UOA)

SOA is an Information Technology (IT) architecture increasingly popular in the telecom industry because it enables the creation of applications via the combination of loosely coupled

\(^2\)capital expenditure, meaning, initial cost/investment  
\(^3\)operational expenditure, meaning, operational cost after deployment
and interoperable services [9]. Typical SOAs make use of Web Service Definition Language (WSDL), Business Process Execution Language (BPEL), Web 2.0 and Mashups. However, the relationship between IMS and the SOA is mostly due to the fact that IMS (like the pre-IMS network(s)) can exhibit web services to an application layer implemented around a SOA.

On the other hand a UOA allows full advantage of the SIP protocol features. The concept of “event package” present in SIP can be used to distribute information and event notifications in a SIP network (e.g., from Presence to User’s keyboard actions). Also combining SIP and any other protocols, e.g., selecting an appropriate user plane media server, redirecting a User to a web page (e.g., a service portal) or retrieving large volume data, is a privileged way to serve Users depending on their content and delivery needs [10].

For these reasons IMS is seen by many as a solution for network operators to be “more than bit pipes”, as explained in an eponymous article [11]. Services will be created faster stimulating innovation. Possible future services on IMS networks that were not already mentioned are for instance, Push to Talk over Cellular (PoC), Instant Messenger (IM), mobile gaming or a combination of existing and/or future services.

After elaborating on IMS’s objectives and motivation, the main IMS modules will now be briefly explained. The architecture is depicted in Figure 2.3.

![Figure 2.3: IMS Architecture, adapted from source](image)

The Control Layer is made up of several service subsystems and is used for session and media control. The most relevant Control Layer’s functional components to understand NEC’s SDP project, as is explained in Section 4.3, are the Call Session Control Functions (CSCF) and...
2. State of the Art

the Home Subscriber Server (HSS), which together make what is known as IMS Core.

The CSCFs can play three different roles: Serving-, Proxy- or Interrogating- Call Session Control Function (S-, P- and I-CSCF) [5]. The S-CSCF is the proxy server that controls the communication session. It invokes the Applications Servers respective to each requested service and is always located in the home network. The P-CSCF is the IMS contact point for the SIP user agents. The I-CSCF provides a gateway to other domains. It is used for topology hiding or if several S-CSCF are located in the same domain.

The HSS is a secure data-base that stores subscription-related information (subscriber profiles), performs authentication and authorization of the User, and can provide information about the subscriber’s location and IP information.

The Service Layer consists of Application Server (AS)s that host the IMS services (e.g., PoC, IM, mobile gaming).

2.3 Open Mobile Alliance Converged IP Messaging (OMA CPM)

The OMA is an industry forum for developing market driven, interoperable mobile service enablers. It is currently on the process of standardizing version 1.0 of a new enabler for converged communication services named CPM [13] [14]. Its exact aim is to enable the creation of a variety of interpersonal and interactive multimedia communication services that run on top of an IP network, thus addressing some of the listed SDP and IMS features.

Because the NEC SDP project aims at implementing this CPM Enabler as close as possible to the standards it will be briefly explained in this subchapter.

The name of this enabler may be misleading, as it seems to imply that CPM is a pure messaging service, while it is not. In fact, CPM can be defined as a composite specification addressing three service concerns:

- text and/or multimedia messaging enabled services (e.g., SMS, Instant Messaging and Presence Service (IMPS), SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) IM, Email, MMS);
- voice-enabled services (e.g., PoC Voice over IP (VoIP)); and
- video-enabled services (e.g., Video telephony, webcam, Video streaming).

2.3.1 OMA CPM User Preferences Profile (UPP)

CPM, as an international project in the OMA standards, is currently the only driving source where user preferences are being proposed for future services. These proposals will be addressed in this Section. Unfortunately the debate on UPP is still in an initial status with no agreed working draft, only a few individual proposals by OMA CPM members, some of them listed in the
requirements for version 2.1 of the XDM Enabler [15]. For the remaining recommendations no document is currently available to the open public, only to OMA members, so no other citation can be made.

The key word “SHALL” in this Section is to be interpreted as described in [16]. Requirements proposed so far by CPM to UPP are:

- **CPM-HLF-002** – The CPM enabler SHALL provide the User with a mechanism to set preferences based on: his addresses, his devices, the message type, the Media Types, the message priority.

- **CPM-HLF-012** – The CPM enabler SHALL be able to reject a Message or a Session Invitation based on the recipient User’s preferences, e.g. originator address (blacklist), undisclosed sender identity, or message type/content.

- **CPM-HLF-013** – The User SHALL be able to set and manage his preferences within multiple UPPs. Meaning, the User is allowed to have several UPPs, change the contents of each of these profiles at any time and name them according to his convenience.

- **CPM-HLF-014** – For each of his devices, the User SHALL be able to indicate one of his multiple UPPs as the active profile, even if the profile was created using a different device.

- **CPM-HLF-014a** – The User SHALL be able to indicate one of the multiple UPPs as the active profile for address and device combinations.

- **CPM-HLF-015** – The CPM enabler SHALL allow the User to set his UPP on various scopes of settings. For example:
  - settings applying to all the devices that he chooses;
  - individual settings per device; and
  - per contact or category of contacts.

  For example the User can say that he prefers to communicate in audio mode with professional contacts in the “Work” UPP and in messaging mode with the same contacts in the “Home” UPP.

- **CPM-PRS-003** – The CPM Enabler SHALL be able to subscribe/unsubscribe a User to one of his contacts’ presence according to his active UPP meaning the User can choose if he wants to share his presence information (e.g. online, away, offline, etc) and who with.

A basic set of user preferences has also been suggested:

- Communication preferences (i.e. text, voice, video, etc);

- What to do with new CPM Messages and Invitations;
2. State of the Art

- Message delivery method
  - Discard and send notification;
  - Defer delivery of message;
  - Deliver entire message immediately;
  - Store in network based storage;
  - Nothing.

- When and what to store in network based storage;
- When and what to synchronize from network based storage:
  - Message and media;
  - Address book;

and with which of the user devices.

The User can define the preferences in a UPP based on a per device basis, per address basis or per context basis. By context it is meant the situation the User is in, e.g. the User may want to defer delivery of messages while he is in a meeting but otherwise receive the messages immediately, for this the User has to change preferences for delivery method of messages from “defer delivery during meeting” to “immediate delivery outside meeting”.

All of the user preferences are to be taken into account by the service provider while providing services to his subscribers.

There are also a few recommendations, most likely going to be accepted according to this Thesis NEC supervisor, Anett Schülke, concerning storage and handling of profiles. These are:

- To store the multiple UPPs available to one User in an XDMS (see Section 2.5.1) in XML (see Section 3.4.1) format;
- To use XCAP (see Section 2.5.2) as the access protocol for creation/personalization/deletion of the profiles; and
- To store the active UPP name(s) in the network (in case there are several devices registered to the same User, there can be one or several active UPPs).

2.4 Profiling Technologies

The following subsections will address the state-of-the-art technologies and proposals on how to define a format for profile data for interchange among different entities.

Because the NEC SDP project aims at implementing the OMA CPM Enabler as close as possible to the standards, it was soon decided to try to do the same with this Thesis project. Not only to
enable the smoothest integration possible of this Thesis project but also because OMA is the only consortium proposing user preferences for future services. This choice is much better explained in Section 5.1 where it is compared with the alternative profiles Composite Capabilities/Preferences Profile (CC/PP) and Generic User Profile (GUP).

Before addressing user preferences the CPM working group created a Shared User Profile (SUP) which is of particular interest to this Thesis project because initial proposals by CPM for the UPP suggest that both the same logical repository for profile documents (XDMS) and the same protocol for access and manipulation of such documents (XCAP) will be used. For this reason, the candidate version of the SUP provides a very good source of information on how to implement a UPP despite addressing a different motivation and purpose, and will thus be explained in this Section.

The alternative profiles CC/PP from World Wide Web Consortium (W3C) and GUP from 3GPP are described afterwards, although they were studied as backup solutions in the event it was impossible or impractical to implement the OMA CPM suggestions for a UPP.

### 2.4.1 OMA CPM Shared User Profile (SUP)

The OMA is driving for standardization the SUP version 1.0 [17] for their CPM and XDM Enabler (see Section 2.5.1).

The purpose of the SUP is to store user information in the network so that Users/ASs can search and find Users according to a certain criteria, thus improving relations. This would allow e.g. search queries to discover communication partners or requests to obtain information regarding a specific User.

The Shared Profile XDMS is the logical repository for SUP documents. It is a server that supports the following functions:

- Stores SUPs in XML format;
- Enforces authorization policies on incoming SIP and XCAP requests;
- Uses the SIP subscription/notification mechanism to notify subscribers of changes in XML documents;
- Provides aggregation of notifications of changes to multiple XML documents; and
- Provides search results.

XCAP is the protocol used to access and manipulate any profile in an XDMS. Both XDM and XCAP will be explained in Sections 2.5.1 and 2.5.2 respectively.

The key words “SHALL” and “MAY” in this Section are to be interpreted as described in [16]. The SUP content is organized under two different sub-profiles:
2. State of the Art

- The User Profile is meant for publicly accessible information for search purposes.
  
  The User Profile Application Usage MAY support search. If the search feature is supported, it SHALL be possible to search for Users based on the data stored in their respective User Profiles.
  
  The main authorization policies for manipulating a User Profile are:
  
  - Users SHALL have permission to read from and write on their own User Profile;
  - Users SHALL have permission to perform retrieve operations of any User Profile in the Users Tree of the XDMS (for search purposes);
  - Users SHALL have permission to perform subscribing to changes operations of any User Profile in the Users Tree; and
  - Users SHALL have permission to perform search operations of any collection of User Profile in the Users Tree.
  
- The Locked User Profile is meant for information the User should not be allowed to edit (e.g. birth date to control adult only content).
  
  As Authorization policies:
  
  - the Service Provider SHALL be the only entity allowed to create this profile on behalf of the User;
  - the Service Provider SHALL have all permissions on the profile; and
  - the User SHALL only have the read permission to this profile.
  
OMA recommends a structure for the XML Schema that describes these two sub-profiles [18]. According to this XML Schema and despite having different access policy rules, the XML elements detailing the User Profile and the Locked User Profile are under the same XML element called <user-profile>, and not separated in two as one might think.

So being and regarding the User Profile, the <user-profile> element in an XML document:

- SHALL include a “uri” attribute that contains the XUI (XCAP User Identifier) of the User for whom this User Profile is intended;
- MAY include any other attribute for the purposes of extensibility;
- MAY include a <communication-addresses> element containing a list of elements representing the communication associated with the User. These elements MAY be of the following kind:
  - SIP URI [6];
  - telephone URI [19];
2.4 Profiling Technologies

- E.164 number; or
- email address.

- MAY include a <display-name> element containing a suggested name to display in user interfaces (e.g. in the IM buddy list);

- MAY include a <birth-date> element containing the birth date of the User;

- MAY include a <name> element containing the human identity of the User. It MAY contain:
  - a <given-name> element;
  - a <family-name> element;
  - a <middle-name> element;
  - a <name-suffix> element;
  - a <name-prefix> element; and/or
  - any other elements from any other namespaces for the purpose of extensibility.

- MAY include an <address> element containing a postal address of the User. It MAY contain:
  - a <country> element corresponding to the country in which this address is located;
  - a <region> element corresponding to the region (e.g. state, province) in which this address is located;
  - a <locality> element which represents the locality in which this address is located (e.g. village, city, town);
  - an <area> element which represents the subdivision of the locality in which this address is located (e.g. neighborhood, suburb, district);
  - a <street-name> element which represents the name of the street in which this address is located;
  - a <street-number> element which represents the house number in the street in which this address is located;
  - a <postal-code> element which represents the code for postal delivery (e.g. ZIP code) for this address; and/or
  - any other elements from any other namespaces for the purpose of extensibility.

- MAY include a <gender> element, containing the gender of the User;

- MAY include a <freetext> element containing a description of the User;

- MAY include a <communication-types> element containing a list of the communication abilities of the User for human consumption;
2. State of the Art

- MAY include a `<hobbies>` element listing the User’s hobbies;
- MAY include a `<favorite-links>` element listing the User’s favorite links; and
- MAY include any other elements from any other namespaces for the purposes of extensibility.

Regarding the Locked User Profile, the same `<user-profile>` element:

- SHALL include a `<birth-date>` element containing the birth date of the User; and
- MAY include any other elements from any other namespaces for the purposes of extensibility.

2.4.2 W3C Composite Capabilities/Preferences Profile (CC/PP)

The CC/PP [20] is a specification for defining device capabilities and user preferences of client devices. CC/PP is a vocabulary extension of the Resource Description Framework (RDF) [21], both designed by W3C. RDF is a metadata data model mainly used to provide interoperability between applications that exchange Web resources. CC/PP is not a specific vocabulary for specifying device capabilities or user preferences. Rather it is a generic language for constructing such vocabularies. It allows interoperability, in part because the data is human-readable enabling it to be interpreted not only by local computers but also by independent parties.

Let’s examine the example snippet in the Figure 2.4

```xml
<xml version="1.0"/>
  <ccpp:component>
    <rdf:Description rdf:about="http://www.netlab.nec.de/myprofile#MyProfile">
      <ccpp:component>
        <rdf:Bag>
          <rdf:li><exa:name>Personal Mobile</exa:name></rdf:li>
          <exa:name>friends</exa:name>
        </rdf:Bag>
      </ccpp:component>
      <ccpp:component>
        <rdf:Bag>
          <rdf:li><exa:name>Work Mobile</exa:name></rdf:li>
          <exa:name>work</exa:name>
        </rdf:Bag>
      </ccpp:component>
    </rdf:Description>
  </ccpp:component>
</rdf:RDF>
```

Figure 2.4: CC/PP example code
The example snippet shows how RDF and by consequence CC/PP makes use of the advantage offered by its XML encoding for its interchange syntax and, as is seen in line number 2, the XML namespace framework for its facility to precisely link each property with the schema that defines it, thus enabling extensibility. Also, the CC/PP example snippet lists two example UPPs consisting of the UPP name (lines 8 and 19) and which contact groups (lines 11, 12 and 22) are allowed to establish a communication session with the device.

2.4.3 3GPP Generic User Profile (GUP)

GUP is 3GPP’s standard in user profiling for all user-related data including services he has subscribed to, user equipment, etc. GUP is a conceptual description that provides an architecture, interface mechanisms and data description to enable the usage of user-related information.

The architecture is depicted in Figure 2.5. The GUP Server authenticates access to data and simplifies data retrieval, by giving a single point of access and hiding actual data locations. The GUP Server after granting access authorization to applications, can work either as a proxy server by directing the request to the appropriate Repository Access Function (RAF) (s), and consequently GUP Data Repository(s), or as a redirect server by returning to the application the address of RAF(s) allowing it to directly request the data from RAF(s).

The RAF realizes the harmonized access interface to the GUP Data Repositories, by hiding the implementation details of these, storing standard formats for data retrieval and performing protocol and data transformation where needed.
2. State of the Art

The storage format and the interface between the RAF and the GUP Data Repositories are not specified by the GUP standards and depend on implementation. The RAF GUP Server and GUP Data Repositories usually reside within the same network.

The Rp interface establishes the connection between both the GUP Server and the RAFs. Applications can also use the Rp interface for direct access to the RAFs, without the help of the GUP Server. The Rg interface allows applications to connect to the GUP server and create, read, modify and delete any data profile using defined GUP methods, which are independent of where the data is located.

Communication between GUP entities is performed via Simple Object Access Protocol (SOAP) / XML messages.

2.5 Implementation Alternatives for User Profiles

As mentioned in Section 2.4 it was decided to try to follow OMA CPM suggestions for UPP. This Section will thereby elaborate on the technology recommended to store and manage these profiles, XDM, and on its access protocol, XCAP. A MySQL database accessed via PHP: Hypertext Preprocessor (PHP) is also detailed as a backup solution.

2.5.1 OMA XML Document Manager (XDM)

The OMA has an XDM Enabler currently with a candidate specification, architecture and requirements version 2.0 [25], 2.0 [26] and 2.1 [15] respectively. As mentioned in Section 2.4.1, the OMA is working on a proposal for a UPP for XDM version 2.1. Besides the OMA XDM has also been endorsed by external organizations/initiatives such as TISPAN and PacketCable.

The XDM Enabler specifies that user-specific service-related information be stored in logical repositories in the network, called XML Document Management Server (XDMS)s, where it can be located, accessed and manipulated (e.g., created, changed, deleted, etc.), how such information is defined in well-structured XML documents, as well as the common protocol for access and manipulation of such XML documents (XCAP).

The XDM Specification defines the following protocols that Users can use through XML Document Management Client (XDMC)s to interact with the Enabler:

- The common protocol, XCAP [27], by which XDMC}s can store and manipulate their service-related data (e.g. UPP)s, stored in the XDMS as XML documents. Each XDMS is defined by an XCAP Application Usage (explained in Section 2.5.2) which performs its functions based on its configuration.

---

4 PacketCable is a CableLabs-led initiative to develop interoperable interface specifications for delivering advanced, real-time multimedia services over two-way cable plant.
2.5 Implementation Alternatives for User Profiles

- The **SIP** subscription/notification mechanism by which Users can be notified of changes to **XML** documents. There is a document called **XCAP diff** [28] (short for difference) that defines a data format which can convey the fact that an **XML** document managed by **XDM** has changed. This way an **XDMC** can send a **SIP** SUBSCRIBE request to the **XDMS** for the “xcap-diff” event of an **XML** document; and if the document changes, receive a **SIP** NOTIFY request with this “xcap-diff” document indicating how the document changed. This **SIP** event notification framework allows clients to be synchronized with documents in the server.

- The mechanism by which Users can search service-related data stored as **XML** documents in the **XDMS** using limited XQuery over Hypertext Transfer Protocol (HTTP). Such search requests as target a single Application Usage (e.g. User Profile) but the scope can range from a single document to all documents of all Users.

Figure 2.6 depicts the **XDM** architecture.

---

**XDM**s major functional entities are:

- the XML Document Management Client (**XDMC**) which can accesses various **XDMS** features. It can be implemented in both terminal (e.g. user device) and server (e.g. co-located with an **AS**) entities. **HTTP** Digest is used for authentication in the **XDMC** and **XDMS** while...
Transport Layer Security (TLS) is used for integrity and confidentiality protection to the exchanged messages in XCAP.

- the Aggregation Proxy that acts as single point of contact for XCAP & Search Requests from terminal-based XDMCs (network-based XDMCs directly access XDMS). It routes requests to appropriate network entity(s) and aggregates responses;
- the Shared XDMS is a logical repository for XML documents that can be reused by multiple Enablers. Enabler-specific XDMS are defined by the specific Enabler (e.g., Presence XDMS in the SIMPLE Presence Enabler); and
- the Search Proxy routes Search Requests to appropriate network entity and aggregates responses;

Each XML document stored in an XDMS is described as an XCAP Application Usage, which enables applications to use the document via XCAP. XDM defines six Application Usages which can be re-used by multiple Enablers:

- Shared Group XDMS - Group
- Shared Policy XDMS - User Access Policy
- Shared Profile XDMS - User Profile
- Shared Profile XDMS - Locked User Profile
- Shared List XDMS - URI List
- Shared List XDMS - Group Usage List

The Application Usages of more importance to our scope are the two Shared Profile XDMS profiles (which were explained in Section 2.4.1). For detailed information on Application Usages check Section 2.5.2.

The SIP/IP Core is the most important external functional entity providing services to the XDM Enabler. It is a network of servers, such as proxies and/or registrars, which, if realized within an IMS architecture, performs the following functions:

- Routes the SIP signaling between XDMCs and XDMSs;
- Provides discovery and address resolution services;
- Supports SIP compression;
- Maintains the registration state; and
- Provides charging information.
2.5 Implementation Alternatives for User Profiles

2.5.2 XML Configuration Access Protocol (XCAP)

XCAP is an access protocol developed by the Internet Engineering Task Force (IETF SIMPLE working group, having become a standard in May 2007 [27]. Its purpose is for clients to be able to retrieve, delete and insert pieces of hierarchically organized data stored in an XML format on an XDMS.

XCAP defines:

• A set of conventions for mapping XML documents, and its hierarchically arranged data, in an XDMS into HTTP Uniform Resource Identifier (URI)s. These HTTP URIs can represent any piece of data and so being interpretation of the URI is left to the XDMS. Because of this HTTP URI structure, normal HTTP primitives can be used to manipulate the data:
  – GET - retrieve content;
  – PUT - place content. Modifying and adding use this command, the difference comes simply whether or not the element already exists, if yes it’s modified, if not it’s added;
  – POST - pass data to a process;
  – HEAD - get meta-data, not content;
  – OPTIONS - query server for capabilities; and
  – DELETE - remove a resource from a server.

• The concept and structure of an Application Usage by which XML documents can be described.

Each XDMS has an Application Usage to define how it can be used with XCAP. The key components of an Application Usage are:

• An AUID (Application Unique IDentifier) to identify the different Application Usages;

• Data Validation Constraints, i.e. an XML Schema, are stored in the XDMS so that the XML documents being inserted have the elements, attribute types, etc in the desired format. Check Sections 3.4.1 and 3.4.2 for details on XML files and XML Schemas respectively;

• Resource interdependency. Rules are also stored in the XDMS on how an operation on one element may affect other elements; especially cross-document affection. These are important for presence related information for example, i.e. IM-like online, offline, away, etc, status information; and

• Authorization policies associated with access to the data. These are application specific; however, XCAP defines a “default” in which a User can read and write their own data and global data is readable by everyone, but writable only by privileged Users.
2. State of the Art

The [URI] construction is divided into three parts:

- The [XCAP] root that identifies the start of [XCAP] tree and under which all resources exist, e.g. “http://xcap.example.com” for domain “example.com”. Both the next two parts, the Document and Node Selector, work just like a UNIX file system path;

- The Document Selector, which is based on a mandatory hierarchy:
  - [AUID];
  - “users” or “global”; “users” are for per-user documents, “global” are for data that is not user specific, i.e. for reading by all Users of the application; and
  - Within Users, next is the username, and next the document name.

For example, to access “doc1” in Figure 2.7:

```
http://xcap.example.com
```

```
user-profiles
```

```
users
```

```
alice
```

```
bob
doc1
dir1
```

```
global
```

```

Figure 2.7: XCAP root and Document Selector schema example
```

the [URI] would be “http://xcap.example.com/resource-lists/users/bob/doc1”; and

- The Node Selector that is used to choose the [XML] component (element or attribute) inside the [XML] document. This is done by navigating the [XML] tree. For example, in the code snippet presented next:

```
<neighborhood>
  <house owner="alice" url="sip:alice@example.com">
    <rooms>2</rooms>
  </house>
</neighborhood>
```
2.5 Implementation Alternatives for User Profiles

The sequence
```
~/neighborhood/house/rooms/
```
would return the element in line 4:

```
<rooms>2</rooms>
```

To select an attribute it’s necessary to prefix its name with the caracter “@”. In the code snippet presented next:

```
<?xml version="1.0" encoding="UTF-8"?>
<foo>
  <bar attr="1">Hi</bar>
  <bar attr="2" bool="y">How</bar>
  <movie stuff="LOTR">Are</movie>
</foo>
```

the sequence:

- foo/bar[@attr="1"] would return the element in line 3: `<bar attr="1">Hi</bar>`;
- foo/bar[@attr="2"]/@bool would return “y”; and
- foo/movie/@stuff would return “LOTR”.

Regarding simultaneous multiple modifications, XCAP provides no lock but uses entity tags (ETag) borrowed from the HTTP protocol. An ETag is basically a version identifier for a resource assigned by the server, that changes every time the resource changes.

For all these reasons XCAP is meant to help leverage the technologies (widely used in phones, PCs and other client devices) of how profiles are stored, accessed and monitored.

2.5.3 MySQL and PHP

Storing the UPP in a Relational Database Management System (RDBMS) under the form of tables (as well as the relationship between UPPs) and accessing it via a Web based interface is a backup alternative to XDM and XCAP that was considered during the development of this Thesis and will be explained in this Section.

MySQL [29] is a popular RDBMS that uses Structured Query Language (SQL) as an interface. This popularity is closely tied to the popularity of web applications based on PHP which is often combined with MySQL. MySQL advantages are:

- high portability. MySQL Server works on practically any current system platform;
- high compatibility. All major programming languages (e.g. PHP) with language-specific APIs include libraries for accessing MySQL databases;
- good performance and stability;
2. State of the Art

- unrequiring in terms of hardware resources;
- very easy to use;
- Users can install MySQL Server as free software under the GNU General Public License; and
- supports standard and advanced SQL.

SQL provides the procedural language needed to specify sequences of operations necessary to obtain answers from RDBMS like MySQL. As mentioned SQL is a very important interface to use with MySQL. It covers two functions:

- Data Definition Language (DDL). There are SQL statements that allow the description of data or information structures so that these can be created, deleted and altered, and the result stored in the DataBase. The SQL statements are respectively CREATE, DROP and ALTER; and
- Data Manipulation Language (DML). SQL Allows the querying, insertion, deletion and update data in the DataBase through the SQL statements SELECT, INSERT, DELETE and UPDATE.

A very important feature of SQL is that it can be embedded in another language code, i.e. PHP. PHP, currently in version 5.2.11, is a general-purpose server-side scripting language that is especially suited for web development, for its ability to produce dynamic web pages. PHP can be embedded into HyperText Markup Language (HTML) to create web pages, and these can be dynamic thanks to its ability to use SQL to retrieve data from RDBMS such as MySQL. Just like with MySQL, PHP can be run on many different system platforms and is compatible with almost all Web servers used today. It’s also free software under the PHP License.

PHP is also an object-oriented language like C++ or Java for instance. It includes private and protected member variables and methods, abstract and final classes as well as methods. There is also a standard way to declare constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model. Interfaces and the possibility to implement multiple interfaces are also present.

For all the reasons listed a MySQL Server acting as a server to store the UPP’s and a PHP based Web interface would be a good alternative to XDM and XCAP.
3
Technological Environment

Contents

3.1 OpenIMS ................................................................. 28
3.2 Mobicents ............................................................... 29
3.3 XDM Servers .......................................................... 30
3.4 Java EE 5 ................................................................. 32
3. Technological Environment

This chapter will give an overall explanation of the main architectures used in the NEC’s SDP project, the OpenIMS and the Mobicents. The alternative implementations of an XDMS, Mobicents XDMS and OpenXCAP, considered in the present Thesis are also detailed. Finally the Java EE 5 technologies XML, XML Schema and JavaServer Pages (JSP) used in this Thesis are succinctly described.

3.1 OpenIMS

There are currently several open source projects on the individual components of the IMS architecture but only OpenIMS focuses on enough of them to be considered a reference in the IMS R&D world. OpenIMS is an open source project from the Fraunhofer Institute FOKUS launched on November 16th 2006, implementing CSCFs and HSS, which together form the core elements of all NGN architectures as specified today within 3GPP, 3GPP2, ETSI TISPAN and the PacketCable initiative. The aim of the project is to provide a non commercial IMS implementation which follows the specification, by actively collaborating with the FOKUS partners and by deploying and operating components from own development.

It was used in NEC’s SDP project, as explained in Section 4.3 because it provides the needed testbed for prototyping and proof-of-concept implementation.

The Open IMS Core forms the heart of the Open IMS Playground @ FOKUS as it is depicted in Figure 3.1.

---

PacketCable is a CableLabs-led initiative to develop interoperable interface specifications for delivering advanced, real-time multimedia services over two-way cable plant
The CSCFs implement the Serving, Proxy and Interrogating functions, as described in Section 2.2. These were developed at FOKUS as extensions to the SIP Express Router (SER), which acts as SIP registrar, proxy or redirect server. The FOKUS HSS (FHoSS) was built to manage user data and associated routing rules. It is mostly a configurator for the Database Management System and the interfaces to the CSCFs and IMS application layer.

An external SIP AS is necessary to run the OpenIMS testbed. In the NEC SDP project, Mobicents was used for this.

OpenIMS does not list hardware requirements to be run, only that a current Linux desktop class machine should be enough. So, OpenIMS was run by the NEC SDP team in a Virtual Machine with 512MB RAM and Ubuntu Linux 6.10 on a 3 GHz Pentium 4 with 2GB of RAM and Windows XP.

## 3.2 Mobicents

Mobicents is an open source VoIP platform currently in stable version 1.2.1.GA \[32\] \[33\] and made available for free under the GNU General Public License. It is certified for Java Service Logic Execution Environment (JSLEE) 1.0 \[34\] compliance and some of the JSLEE 1.1 \[35\] features. Mobicents basically serves as a high-performance core for SDP and IMS by leveraging Java EE technologies (part of which will be discussed in Section 3.4) to enable the convergence of data and video in NGN applications.

Because of its modular and extensible nature Mobicents was used in NEC’s SDP project and in this Thesis as will be explained in Sections 4.3 and 5.3.3 respectively. To understand how it was used a brief comprehension of Service Building Blocks (SBB) is necessary.

Mobicents enables the composition of SBB i.e. software components that send and receive events and perform computational logic based on the receipt of events and current state, for Call Control, Billing, User-Provisioning, Administration, Presence-Sensing and other features; they have many similarities to Enterprise JavaBeans (EJB)’s. Furthermore, Resource Adapter (RA) are used by SBBs to interact with external resources via standard or proprietary protocols, e.g. SIP and HTTP.

Mobicents high level architecture is shown in Figure 3.2.

![Mobicents High Level Diagram](source:33)
3. Technological Environment

The main core capabilities used in NEC’s SDP project are:

- The Mobicents JSLEE. It is an AS with an SBB component model and fault tolerant execution environment. External communication to this domain can only be made through RAs.
- The Mobicents Sip Servlets. These are used to deliver an open source platform on which to develop and deploy portable and distributable SIP and Converged Java EE services.
- The Mobicents Presence Service implements, among others, an XDMS (explained in the next Section 3.3.1).
- The JBoss Microcontainer is a hosting environment where higher level containers reside. It provides service registration, configuration and dependency management, class loading isolation control, package versioning, deployment, thread pooling and other fundamental building blocks necessary for scalable, fault tolerant middleware servers.

Mobicents does not list hardware requirements to be run, so, it was run by the NEC SDP team in a Virtual Machine with 512MB RAM and Windows XP on a 3 GHz Pentium 4 with 2GB of RAM and Windows XP.

3.3 XDM Servers

Two open source implementations were considered for the XDMS, the Mobicents XDMS and the OpenXCAP. Both servers possess the same basic functional elements required for this Thesis:

- A Data Source to store user XML documents and server-related information; and
- An Aggregation Proxy to handle XCAP requests from XDM client’s, including authentication and authorization.

The next two sections will elaborate on the particularities of each implementation.

3.3.1 Mobicents XDM Server

Formerly known as OpenXDM, the Mobicents XDMS is part of the Mobicents SIP Presence Service and implements the OMA XDM version 1.1 specification.

In contrast to OpenXCAP, Mobicents XDMS possesses a fully implemented XDM Event Subscription Control, that uses the SIP protocol to handle subscriptions to documents managed by the XDMS. This element has as functions the authentication and authorization of a subscription, the attachment to update events on specific documents or application usages, and the sending of notifications when documents change.

Mobicents includes, parallel to the XDMS, the following Client-Side components:
3.3 XDM Servers

- An **XCAP** Client that internally uses the Apache **HTTP** Client to interact with an **XCAP** Server;

- An **XDMC SBB** which is an interface of a **JSLEE SBB** to be used as a client for an **XDMS**. There is an internal implementation of this interface to be used on applications running in the Mobicents **XDMS JSLEE** container, and an external intended to be used on applications running in a different **JSLEE** container, e.g. OpenXCAP.

- The latter implementation requires an **XCAP** Client **RA** to adapt the **XCAP** Client API into **JSLEE** domain so it can communicate with an **XDMS** external to Mobicents, e.g. OpenXCAP. It provides methods to interact with an **XDMS** in both synchronous and asynchronous ways;

Mobicents also has plans, although not available yet, to implement a User Profile Enabler **SBB** in accordance to the same **OMA CPM** standards for **UPP** this Thesis follows.

### 3.3.2 OpenXCAP

OpenXCAP [37] is an open source **XDMS** provided under the GNU General Public License version 2, that works out of the box with OpenSIPS [38] Proxy/Registrar/Presence servers. It uses storage based on a MySQL server.

The server supports as security features **TLS** encryption and digital certificates using GnuTLS library, digest or basic **HTTP** authentication. There is also the possibility for database passwords to be stored in an encrypted format.

The specific settings for an installation must be set from the configuration file (check Annex A for the configuration file used in this Thesis) which is split in several configuration sections:

- The **XCAP** root is the context that contains all the documents across all applications and Users that are managed by the server.

- There are separate configuration settings for each backend. The current supported backends are Database and OpenSIPS:
  - The “Database” option was used since it contains the **URI** to the MySql database engine, where User registries and **XCAP** documents are kept.
  - The “OpenSIPS” section, besides the same options as “DataBase”, contains the Uniform Resource Locator **URL** where OpenSIPS’s management interface is listening for commands, providing a way to use **SIP** to subscribe to changes on documents for presence related usage.

- The **TLS** section configures the use of **TLS** for integrity and confidentiality protection of exchanged messages.
3. Technological Environment

3.4 Java EE 5

Short for Java Platform, Enterprise Edition version 5, Java EE 5 is a widely used platform for server programming in the Java programming language with emphasis on SOA and next-generation web applications.

Java EE includes many components of the Java Platform, Standard Edition (Java SE) among which of interest to NEC’s SDP project are the open source Java Development Kit (JDK) from Sun technologies, the security model to protect data in both Web and locally based applications and the portability ensured by the Write Once Run Anywhere technology.

The next subchapters will address Java EE 5 new components of more interest to this Thesis: XML, XML Schema and JSP.

3.4.1 Extensible Markup Language (XML)

XML is a W3C Recommendation currently in the fifth edition and a core Java EE 5 technology. XML is a set of rules for encoding documents electronically and has been designed for ease of implementation and for interoperability with HTML.

XML is a markup language that is fast becoming the standard for data interchange and storage on the web. XML is of crucial importance for this Thesis because the proposals by OMA CPM members for the UPP suggest XML be used as the storage and exchange format.

The basic building parts of an XML vocabulary are elements and attributes. An element is constituted by start- and end-tags, attributes and if they are containers, other elements or text. An attribute is a name/value pair that exists within the start-tag of an element.

The main characteristics of XML are:

- it is text based and supports Unicode so it can represent text of almost every human written language;
- it can represent the relevant data structures of computing: lists, registers and trees;
- its own format (e.g. XML Schema) can be used to describe its structure and field names as well as accepted values;
- the restricted syntax and parsing requirements make analysis algorithms more efficient;
- high portability because since it doesn’t depend on hardware or software platforms, one entity can always read XML written by a different entity.
- is editable on various levels of complexity, from basic txt editors to complex software with validation and syntactic analysis; and
- high extensibility. XML namespaces allow the creation of compound documents from several separate sources. A different prefix is added to the elements and attributes of each source,
so as to prevent ambiguity between identically named elements or attributes in the imported vocabulary.

### 3.4.2 XML Schema

XML Schema is a W3C Recommendation with the latest version dating from 28 October 2004.[42][43]

An XML Schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and field names as well as accepted values of documents of that XML type, above and beyond the basic syntactical constraints imposed by XML itself.[44] An XML Schema can this way be used to validate an XML document of the type the schema defines. This process is called validation.

XML Schemas have arisen due to weaknesses in the traditional Document Type Definition (DTD) used to describe XML document types. For instance it uses a different syntax then the XML it’s defining, it’s difficult to write good DTDs, it’s limited in its descriptive powers, not extensible, doesn’t describe XML as data well and there’s no support for the Namespaces proposal.

Unlike DTD syntax, XML Schema syntax is well-formed XML enabling the User to edit it with the same tools he uses for the XML files he’s defining. It also provides much more control over datatypes and patterns, making it a more attractive language for enforcing strict data entry requirements, and is extensible thanks to XML namespaces.[45]

### 3.4.3 Java Server Pages (JSP)

The JSP is a component of Java EE 5 currently in specification 2.1.[46] JSP is a server side Java technology for developing web pages that include dynamic content.

A JSP page contains static markup, like HTML and XML tags, just like a regular web page, but it also contains special JSP elements that allow the server to insert dynamic content in the page. These JSP elements contain variables based on external information, mostly based on the User, which can be used to change the content of the JSP page. This information can be for example the User’s identity, browser type, information provided by him or selections made.

JSP gathers the most important features found in the alternatives (Microsoft’s Active server Pages, PHP, Macromedia’s ColdFusion and Java servlet template engines like Velocity and FreeMarker)[47]:

- **JSP** supports both scripting and element-based dynamic content and allows programmers to develop custom tag libraries to satisfy application-specific needs;

- **JSP** pages are compiled into executable code the first time they’re requested. This makes the server much more faster if coupled with a persistent Java virtual machine; and
3. Technological Environment

- The use of JSP pages enable a more business-oriented logic. Java web developers can implement the application logic as servlets, EJBs, and custom elements, while page authors craft the specifics of the interface and use the powerful custom elements without having to do any programming.

In addition, JSP has a couple of unique advantages:

- JSP is a specification, not a product. This means vendors can compete with different implementations, leading to better performance and quality. It also leads to a less obvious advantage, namely that when so many companies have invested time and money in the technology, changes to it will be around for a long time, with reasonable assurances that new versions will be backward compatible; with proprietary technology, this is not always a given.

- JSP is an integral part of Java EE 5, which means that JSP can play a part from the simplest to the most complex and demanding applications.
## 4 NEC SDP Project

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Motivation for the NEC SDP project</td>
<td>36</td>
</tr>
<tr>
<td>4.2 Personalized Communication Control</td>
<td>37</td>
</tr>
<tr>
<td>4.3 Project Testbed</td>
<td>39</td>
</tr>
</tbody>
</table>
4. NEC SDP Project

This chapter will elaborate on the specific motivations behind NEC's SDP project, its main module (PCC) features and on its high level implementation. The biggest citation concerning this Section [9] is for obvious reasons for NEC internal use only, although the paper submitted by NEC concerning its PCC module is available to the public [48].

4.1 Motivation for the NEC SDP project

According to the Moriana Operator Guide from September 2008 [4] most operators developing SDP 2.0 with Web 2.0 solutions, are leaning toward a SOA for its easiness to provide web and multimedia services. The SOA however tends to ignore the full capabilities of SIP and IMS for the delivery of services. SIP as a protocol and IMS as a service architecture can optimally integrate with and complement a SOA by bringing a dimension that is currently missing to it and is fundamental: user orientation. The challenge is to converge those in a NGN SDP by exploiting their individual strengths in serving the User's service needs: service oriented as well as user oriented (see Figure 4.1).

![Diagram of SOA and UOA](source)

Figure 4.1: Converging SOA and UOA, source [9]

Adding this motivation to all of the ones presented in the IMS chapter, Section 2.2, we get the following challenges driving the NEC SDP research project:

- To build an Application Layer with an horizontal architecture to enable the usage of a Generic SIP session with full exploitation of SIP and IMS capabilities. This would support multimedia communication as well as content/data services across different IP domains. See Figure

---

36
4.2 Personalized Communication Control

Taking in consideration the challenges presented, the research approach was to develop a Personalized Multimedia Communication Control with user-interaction. The concept and design is closely related to the OMA's CPM Enabler explained in Section 2.3:

- When personalizing the communications for a single User, the session and media control concept of the CPM conversation server as well as the CPM user preferences data component were considered;
- During the development of the PCC Module the standards discussion were continuously being monitored and the conversation server concepts being considered for evaluation in the ongoing R&D project.

The PCC Module architecture follows a strong modularized architecture as depicted in Figure 4.3.

The most important feature of this model necessary to understand this Thesis is Policies. The language selected for Policies was the eXtensible Access Control Markup Language (XACML). Rules that include the user preferences planned in the present Thesis, the user context (i.e. where he is and what he is doing) and the device capabilities are used in the PCC to model Policies that decide the User’s communication establishment and modification results.

Policies can be set individually by the User through the mentioned rules or as defaults by the operator. The policy engine within the PCC provides the decisions on the follow up action to
the session management, e.g. accepting a conversation session or not, directing it to a specific device or split the video and the audio of the session and send it to different devices. In the event more than one Policy is valid, policy correlation algorithms are applied.

Policies are specified for the following situations:

- new incoming or outgoing communication sessions;
- update/redirection of existing communication sessions;
- termination of existing communication sessions; and
- registration of devices.

An important feature of Policies is that they are dynamic, meaning that whenever a policy decision is triggered (e.g. a User wants to establish a communication session or when a User reregisters) the rules will be evaluated newly by the policy engine. This means that Policies reflect changes in the variables that influence them during runtime of the PCC and even during a communication session.
4.3 Project Testbed

In order to develop the PCC module in an independent testing environment, NEC used two supporting open source testbed platforms, the Mobicents and the OpenIMS.

Figure 4.4 depicts NEC’s SDP Prototype Deployment and shows how Mobicents was used as a SLEE/SIP AS connected via a SIP-based interface to the the S-CSCF in the OpenIMS testbed. The PCC module is deployed as an AS module realizing a Back-to-Back User Agent (B2BUA). Every module in the PCC is an SBB within the Mobicents framework with the exception of the Data Layer, which was built using the EJB technology. Communications between SBBs are done through SBB interfaces, between an SBB and an external resource through a RA, and to and from the Data Layer via Java Remote Method Invocation (RMI).

![Figure 4.4: NEC SDP Prototype Deployment, simplified from source 9](image)

To better simulate a telecommunications network it’s necessary to have some Users act as clients of NEC and other Users as clients of competing service providers. For this reason a separate S-CSCF was deployed to serve the latter as is also shown in Figure 4.4. This way these Users can access the network without being registered in the PCC, not making use of its advantages, and thus representing for testing purposes a User from a different service provider.

1 The B2BUA is responsible for handling all SIP signaling between both ends of the call, from call establishment to termination.
## Design and Implementation

### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Profile Format Comparison</td>
<td>42</td>
</tr>
<tr>
<td>5.2 User Profiles</td>
<td>44</td>
</tr>
<tr>
<td>5.3 Implementation</td>
<td>49</td>
</tr>
<tr>
<td>5.4 Demonstration Scenario</td>
<td>55</td>
</tr>
</tbody>
</table>
5. Design and Implementation

This section will give the analyses of the alternative implementations presented in Sections 2 and 3, explain the design made for the UPP file, the architecture of the prototype constructed within the NEC SDP project and the demonstration scenario created.

5.1 Profile Format Comparison

Table 5.1 lists the arguments pro and against the use of each of the technology presented in Section 2.4 for the UPP.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Advantages</th>
<th>Drawbacks</th>
</tr>
</thead>
</table>
| UPP under OMA CPM | • standard representation and architecture is under discussion  
• enhances interoperability  
• possibility of extension  
• possibility of merging different data from different sources | • parsing requires large CPU load and memory overhead  
• verbose |
| CC/PP under W3C | • enhances interoperability but to a lesser extent than UPP  
• possibility of extension  
• possibility of merging different data from different sources | • parsing requires large CPU load and memory overhead  
• verbose |
| GUP under 3GPP | • also enhances interoperability but to a lesser extent than UPP  
• possibility of extension  
• possibility of merging different data from different sources  
• provides management methods for the profiles, authentication and authorization rules, and hides data repositories implementation from the system | • parsing requires large CPU load and memory overhead  
• verbose  
• access protocol to the GUP network not standardized yet  
• no discussion is underway for UPP’s content or supporting architecture  
• no proposal on how servers should be, or how to adapt content on behalf of user devices |

The fact that all three alternatives are based on an XML based format creates a lot of the
same advantages and drawbacks to all. XML namespaces allow extensions to the vocabulary and different data to be merged from different sources, while still remaining complaint to the XML model. This flexibility allows new preferences to be added to the designed XML file later on, but also other user related information (e.g. a SUP), that other applications can use to provide services to the User or the Service Provider. How to make use of user related information is a tremendous research area, e.g. for niche market publicity. This way by using the XML format we can enable the extension of other services and applications without interfering with the present design. XML therefore enables a high level of usability and extensibility.

As drawbacks XML documents can require a very large CPU load and memory overhead to be parsed, especially if a User has many UPPs and consequently a large XML file. Also the fact that XML is verbose doesn’t go well with some of the networks involved in IMS because of their low bit rate (in particular the mobile networks). The UPP proposal by OMA CPM does however minimize these two factors by using XCAP. This protocol allows the access and transfer through the network (and posterior parsing) of a single UPP or preference (which is all that is necessary most of the times) instead of the entire user XML file. This way the load on the network is significantly lowered, as well as the CPU load and memory overhead required to parse the information.

Also because each of the three formats is backed by its respective consortium of technology and telecommunication companies, interoperability is assured at least to a certain point. However, there are currently no standard vocabulary for any user profile being proposed for CC/PP or GUP. That leaves CPM under the OMA as the only consortium proposing user preferences for future services. So only this option maximizes the expected interoperability of the solution built and fits this Thesis requirement of staying as close as possible to models under standardization process (in the absence of already standardized models).

Another important requisite for this Thesis project is to have the smoothest integration possible in the NEC SDP project it is part of. Since one of the main aims of the NEC SDP project was to implement its PCC as close as possible to the OMA CPM Enabler standards, the option to follow the models under standardization for the UPP by the very same working group in OMA would be but a continuation of the said NEC SDP aim. This would also allow the easiest integration path because both the pre-existing in the NEC SDP project and the to be built modules in this Thesis would have been planned and either standardized, or in the process of being so, by the same working group. A good quality of design and a rewarding learning experience from the diverse specifications involved would also be assured.

Taking into account all these aspects, it was decided to follow the recommendation by OMA CPM for how to design the UPP content and build the supporting architecture.
5. Design and Implementation

5.2 User Profiles

For each User it was decided he would have one single file with all his UPPs. But since OMA CPM is very likely to standardize the same storage (XDM) and communication (XCAP) protocols to UPP as the ones from SUP, it was decided to insert the SUP in the same user file as the UPPs. Despite the SUP not being the focus of this Thesis, this decision was taken for extensibility purposes because this way, if in the future NEC decides to implement the SUP, it will be possible to focus just on how to make use of the information contained in the SUP since the work of building the profile per se, the storage and the handling of the profile will have already been done.

The OMA CPM specifications for SUP suggest there be one XML file, with one SUP per user and recommends a structure for an XML Schema to control how these SUP XML files have to be.

For UPP, OMA CPM also suggests there be one XML file per user and an XML Schema be used to control these UPP XML files, but unlike for SUP, there can be more than one UPP per user, and thus per XML file, and OMA CPM does not suggest yet a structure for the said XML Schema file.

As it was decided to join the SUP and UPPs belonging to one user in one single XML file, the OMA CPM suggested XML Schema for SUP and the XML Schema proposed by the author of this Thesis for UPP were joined together in one. This XML Schema file is named “user-profiles.xsd” (thus the name of this Section) and is annexed in Annex C. Next it will be explained how this joint XML Schema organizes the SUP fields and the UPP preferences. The pictures presented are from the program Liquid XML Studio [49] which was used to develop the XML Schema.

Figure 5.1 shows the root element of all profiles, <user-profiles>. A reminder that each User has one XML file with all his information. This information is separated in one <shared-user-profile> element and in one or more <user-preferences-profile> elements. As the name indicates the first is based on the SUP and the second on the UPP. The <shared-user-profile> element is a container for the elements <user-controlled-info> and <locked-info>, based respectively on the sub-profiles User Profile and Locked User Profile of the SUP standard.

![Figure 5.1: Initial levels of depth of the XML Schema](image)

5.2.1 Shared User Profile (SUP)

Figure 5.2 represents the attributes and elements in the <shared-user-profile>.

All of the attributes and elements proposed in the SUP standards and listed in Section 2.4.1 were included. These parameters will not be explained individually because they were already
5.2 User Profiles

(a) XML Schema of the User Controlled Info Sub-Profile

(b) XML Schema of the Locked Info Sub-Profile

Figure 5.2: XML Schema of the Shared User Sub-Profile's

listed in Section 2.4.1 are mostly self-explanatory and SU P is not the focus of this Thesis.

However, it should be noted that the <communication-addresses>, <name>, <gender> and <address> elements were changed from the sub-profile element <user-controlled-info> to the <locked-info>, because it was considered that this information should not be controlled by the User but by the service provider. Two personal additions were also made to these sub-profiles that are:

- To the <user-controlled-info>, the <devices-name> element. This element lists the nicknames chosen by the User for his devices. Two approaches are possible for these nicknames; either the User is free to choose the names or he has to choose from a given set. Although the first option seems more user friendly it comes with a problem, for example a caller wants to call the “business device” of a friend but that friend named it “work device”. This presents a problem of not simple resolution. Translation of these names also poses a problem although of simple solution if the names are predefined. For these two reasons a first approach of a well-defined list was planned with a future approach of real user choice.

- To the <locked-info>, the <publicity-settings> element. Publicity related options are currently under debate but no proposal has been made. A possible and feasible solution in the opinion of this Thesis author is that publicity related settings be set by the service provider depending on the user tariff plan and/or revenue he provides, i.e. if the User has the cheapest plan then he gets more aggressive publicity, while a User providing a higher revenue gets less publicity. The following options How this dependency could be established requires study on user-publicity relations and so it is left for the service providers to decide. None the less several options on how the publicity feeds could be played in user devices
were considered for this parameter:

- straight away (probably unfeasible even for aggressive marketing since it may for example overlap an ongoing audio communication session);
- as soon as the User is not using the device (e.g., when it is turned on or after the User finishes an audio communication session);
- same as previous but asking the User before playing, thus demanding his express permission; and
- never.

5.2.2 User Preferences Profiles (UPP)

Figure 5.3 shows the attributes and elements of the built UPP.

Figure 5.3: XML Schema of the UPP

All the user preferences suggested by the OMA CPM (and listed in Section 2.3.1) were included in the XML Schema. Because the UPP is the focus of this Thesis, unlike SUP all preferences proposed by OMA CPM will be listed and explained next:

- the attribute “profile-name” contains the name of the UPP;
- the element <session-invitation-types-accepted> has the boolean attributes “video”, “video-only-with-wlan”, “sound” and “text”. These are meant to filter what type of communication sessions the User accepts to be initiated with him. Both the User initiating the communication session and the one receiving it should always receive a report of the denied invitation. The sound and video option were implemented in the PCC and thus in the prototype of this Thesis as showed in Section 5.4.
5.2 User Profiles

- the element `<message-delivery-method>` lists options to what to do with an incoming message (not a communication session). Each type of message (video, sound and text) can have one of the next values:
  - `<disregard>`. Deletes message;
  - `<discard-and-send-notification>`. Deletes message but User receives a report of the denied message;
  - `<defer-delivery-of-message>`. Message is stored in device based storage and only played when the User chooses to; or
  - `<store-in-network-based-storage>`. A report is sent to the User warning about the message stored in the network based storage. The User has access it to see the message.

An option to deliver messages immediately was initially considered but removed because the next preference already has this option as will be explained.

- the element `<synchronize-from-network-based-storage>` is a container for two other elements:
  - `<message>`. This element describes whether the User wants to synchronize messages from a network based storage or not, and if yes, the whole message or just the headers (i.e. what type of message it is, who it is from, what is the title of the message, etc) and with which user devices.
    This is meant for the User for example to be able to only receive warnings of new video messages in his mobile device, but download the same messages immediately to the desktop pc device because it has a better resolution to play them.
  - `<address-book>`. This element describes whether the User wants to synchronize the address book containing the User’s contacts from a network based storage, and if yes with which user devices.
    The idea is if he adds a new contact in one of his devices, the contact is added in for instance all of his registered devices, or just in the work ones, or just the personal ones, etc.

The following preferences were added by the author of this Thesis:

- the element `<ringing-loudness>` to control how loud the device should be set. It can be of the type “loud”, “normal” or “quiet”;
- the element `<presence-info>`: Some research was done on Presence information (i.e. IM-like status: online, offline, away, etc) for [IMS] but dropped because the NEC [SDP] project did
not have such feature implemented nor plans to implement it. Still the element was included as a basic proposal of who should have access to the User's Presence information. This is done through its boolean attributes “hide-from-all” and “available-to-all”. The possibility of not showing everyone the user status but just a select set of contacts and/or contact groups is foreseen by the sub element `<available-only-to>`;

- the element `<sessions-while-roaming>` possesses the boolean attributes “video”, “sound” and “text” to contain information on which type of communication sessions the User wants to receive while in roaming abroad;

- the element `<session-invitation-acceptance>` possesses the boolean attributes “all-calls”, “only-urgent-calls”, “only-private-calls” and “no-one” with self-explanatory meanings. Although apparently there is no option in [SIP](https://tools.ietf.org/html/rfc3261) to implement urgent or private communication session it might be possible to implement such camp in the [SDP](https://tools.ietf.org/html/rfc4566) message header. This would have to be investigated when implementing such preference in the NEC SDP project. The sub element `<only-from>` lists the individual contacts and contact groups with permission to contact devices with this UPP active.

This sub element was implemented in the [PCC](https://tools.ietf.org/html/rfc3208) and thus in the prototype of this Thesis as showed in Section 5.4;

- the element `<when-to-activate-session-voice-mail>` lists options on when to have voice mail active. It possesses the self-explanatory boolean attributes “when-device-is-offline”, “when-user-is-busy” and “when-call-is-not-answered”; and

- the boolean attribute “wlan-auto-change” which sets if the user device should automatically change from a GMS or 3G Network to predefined WLANs if they become available.

Joining the setting `<session-invitation-types-accepted>` with `<session-invitation-acceptance>` was considered but disregarded. The idea was to besides choosing what communication type to accept, to also choose who from (e.g., allow receiving video, audio and text sessions from the “family” contact group in the “home” UPP but only sound and text from the “work” contact group). It was considered too complicated to fill-in so many text entries and check boxes in a user interface, considering the limited user satisfaction gain. The [IMS](https://tools.ietf.org/html/rfc4465) feature to renegotiate sessions at start would mitigate this limitation since for example, if the User receives a video and audio communication session from a contact but only wants audio, the User can choose to renegotiate and accept only an audio call.

### 5.2.3 Default UPPs

The author of this Thesis has defined the preferences of a set of default UPPs bearing in mind what a normal User needs and wants on a per device, per context and per address basis. The
5.3 Implementation

idea is to suggest to a future User of the service a set of UPPs he can use, so he can quickly set his UPPs and make use of the service. This way the User can just make some adjustments to the UPPs based on his personal preferences, instead of being forced to fill-in long tables of forms to create all of the UPPs he wants from scratch, which is far from being user friendly. The construction of these UPPs was also made to help the author of this Thesis better imagine what preferences are useful for the User and how these can be organized in a profile.

The name of the UPPs created are listed next:

- Home;
- Work;
- Meeting;
- Driving;
- Sleeping;
- Roaming;
- Public Places;
- Personal Mobile, Personal Desktop and Personal Laptop; and
- Work Mobile, Work Desktop and Work Laptop.

For the preferences contained in each of these UPPs, check Annex B.

5.3 Implementation

As explained in Section 5.1 it was decided to implement the OMA CPM’s proposals on the design of a UPP and its supporting architecture. This architecture was then planned having these proposals in mind but also a good quality of design, a smooth integration path in NEC’s SDP project and making sure the solution was extensible to enable future work to be implemented.

Having all of this in consideration the architecture depicted in Figure 5.4 was developed by the author of this Thesis to support the UPP and interact with the NEC SDP project. The number references in the connections will be used to help explain in Sections 5.3.2 and 5.3.3 the implementation options made.

The modules inside the dashed line were developed and/or deployed by the author of this Thesis and will thus be explained in this Section. The remaining ones in the Figure were already developed by the NEC SDP team, though they required some adjustment to interact with and incorporate the former ones.
5.3.1 XDM Server

To start, a choice had to be made between the two available open source implementations of an XDMS: Mobicents XDM Server or OpenXCAP.

Mobicents XDM Server does have better documentation than OpenXCAP (which possesses virtually none, only some configuration tips), but the decisive factor was that Mobicents XDM Server was not stable enough to be used when this decision had to be made. This was because the open source code behind Mobicents XDM Server had been known as OpenXDM until just one month before; when it was renamed to Mobicents XDM Server and the integration process into the Mobicents project started. Because this change was so recent the code was not yet stable within the Mobicents platform and therefore could not be used. This was rather unfortunate since the NEC SDP team even used the Mobicents platform in its project implementation; but the decision to use OpenXCAP was made (with a future work approach of replacing OpenXCAP with the Mobicents XDM Server after the latter was stabilized).

Adding to this, neither of the XDMS implementations permit an easy implementation of the extension necessary to allow the storage and management of an XML document with a different design then the ones the XDMS was built to store. This is because an entire new XCAP Appl-
5.3 Implementation

cation Usage is necessary, which to implement requires a thorough understanding of both the XDM and XCAP specifications, but more than that, of the specific implementations made of these specifications in these servers; thereby making this task a presumably very heavy time consuming one. However, not to use an XDMS and consequently XCAP would greatly weaken the design value of this Thesis work implementation. So two options arose:

- either to store the XML files designed by the author of this Thesis in the XDMS with the XML Schema validation turned off (which presented potential problems in the later reading of the files in case they were not submitted in a correct form);

- or to “hack” an already implemented Application Usage by replacing its XML Schema by the new User Profiles XML Schema. Unlike with the previous option, this would enable User Profiles XML documents being stored, to be adequately validated by the User Profiles XML Schema.

Obviously the last option is very poor practice, but it does allow prototyping and proof-of-concept which is the aim of this Thesis. So, caused by time and effort constraints the option to “hack” the “resource-lists” Application Usage of OpenXCAP was taken. In practical terms the only difference is that instead of having the XML file for instance for User Bob, in URL:

```
http://xcap.example.com:8000/user-profiles/users/sip:bob@open-ims.test/user-profiles.xml
```

It is located in:

```
http://xcap.example.com:8000/resource-lists/users/sip:bob@open-ims.test/resource-lists.xml
```

The deployment and use of OpenXCAP was rather straightforward, apart from the just mentioned “hack”. The configuration file used is annexed in Annex A and lists the choices made. Notice should be made that TLS encryption and digital certificates were not used since security of implementation is not a main focus in this Thesis, however besides the full access path to the user file, a password still needs to be submitted to access OpenXCAP to make sure each User is only accessing his own user data.

5.3.2 UPP Interface

A user-end interface is necessary in this Thesis for a User to create, edit or delete his UPPs and to test the designed UPP. For this purpose a basic Web Interface based on JSP and HTML was developed in Eclipse [50] that:

- runs in a normal web-browser like Mozilla Firefox [51] in a desktop computer;

- has all options necessary to interact with the XDMS via XCAP and test the implementation; and

- retrieves and presents information from the XDMS.
Figure 5.5: UPP Interface Flowchart

Figure 5.5 represents the flowchart of the UPP Interface.

Each circle in the schematic represents one JSP page, or a group of closely related ones. Each connection represents an action by the User or a logical step between JSP pages.

In simple terms, in the Initial Page a User can login to an existing User, delete a User or create a new one. If he chooses to create a new one and after the chosen display name is checked for uniqueness, he is asked to create the “user-profiles”, which includes the SUP and the UPPs. For this he can either choose:

- to create the entire “user-profiles” from scratch by filling in and submitting an extensive form;
- or

- to use a default “user-profiles” that includes an example SUP and the set of previously created UPPs listed in Section 5.2 and annexed in Annex B.
5.3 Implementation

After either creating a new User or logging in to an existing one, the User gets to the Manage User page. Here he can edit a [UPP] or change the current active [UPP] of one of his devices. Unfortunately if the UPP he is editing is the active one or if he changes the currently active UPP for one of his devices, this edit/change is not being signaled to the PCC because such feature was not implemented for time constraints, thus the partial success note in the connection between the “Confirm change of active UPP” and the “Manage User” page in the previous Figure. For more information on this issue check Section 6.2 under the “Implement SIP from the UPP interface to the Session Management” item.

To serve this Web Interface a infrastructure was built in Java also using the Eclipse software. The classes and methods built main functions can be grouped into:

- Java constructors to build the SUP and the UPPs, when the Web Interface User chooses to create the “user-profiles” from scratch;
- XCAP commands related methods in order to build an XDMC compliant to the XCAP specifications. These methods use the cURL command-line tool to use the HTTP commands PUT to place content in the XDMS, GET to retrieve it and DELETE to delete it (connection 1 in Figure 5.4). This content should have been any part of the XML file containing a “user-profiles” (e.g. element, attribute, etc) but unfortunately the OpenXCAP server was not implemented up to this extent, i.e. an XML file can be inserted, retrieved and deleted as supposed to, but not parts of it;
- Because of this unforeseen problem it was necessary to use the Xerces Java Parser library to create and handle Document Object Model (DOM) tree representations of the XML files. Only by this way individual elements in an XML file (e.g. a UPP, a preference, etc) could be handled; and
- Methods to record the active UPP for each user device, and to which User it belongs to, in the Data Layer (connection 2 in Figure 5.4) and read from it (connection 5). This is done through calling a Java RMI method previously implemented by the NEC SDP team. The active UPPs are recorded in the Data Layer, so that the PCC can access this information when it is turned on. The preferences in each active UPP are also recorded in the Data Layer for backup purposes.
- Several test methods were also built to print various information to logs and/or the screen.

5.3.3 UPP Manager

The UPP Manager was developed as several classes in Java in Eclipse to appropriately link the pre-existing modules in the PCC with the XDMS.
5. Design and Implementation

Because the modules inside the PCC have been built under the form of SBBs in the Mobicents framework (except the Data Layer that was build with the EJB technology), and the only way for interactions to happen between these SBBs and external resources is through RA, it was necessary to choose between to implement the UPP Manager:

- either as an external resource, meaning communications 3 and 5 in Figure 5.4 would have to happen through RA;
- or also as an SBB inside the Mobicents framework. This way only communication 5 in the same Figure would happen via a RA Communication 3 would happen via an SBB interface (for the specifications of these check [35]).

Communications 2 and 4 would happen via Java RMI in either case. So since the Mobicents working group had already implemented an XCAP RA that could be used in 5; and NEC SDP has already built an SBB interface that could be used in 3 (in opposition to building a new RA) it was decided to implement the latter option as the same Figure shows.

Contained in the said SBB the UPP Manager’s main function is to assist the PCC when it is turned on:

1. When this happens, the Conversation Control module needs to know which are the active UPPs for each of the user devices registered to be used, and what preferences are contained in these UPPs. This information is necessary so it can be provided to the Policy Decision Point so that when setting up a communication session for a User, and for its duration, the relevant Policies for the User can reflect the preferences on the active UPP of each of his user devices, i.e. so that the active UPPs can affect the decision of accepting or not the communication session and if yes, to forward it to which of his user devices.

So for this to happen the Conversation Control module asks and later on receives these preferences from the UPP Manager (connection 3 in Figure 5.4).

2. After receiving this request the UPP Manager retrieves from the Data Layer (connection 4 in Figure 5.4) which is the active UPP for each user device and to which User.

3. With this information and using the Mobicents XCAP Client and XCAP RA the UPP Manager retrieves the active UPPs from the XDMS via XCAP (connection 5 in Figure 5.4).

4. As mentioned in Section 5.3.2 because of the limited XCAP features of OpenXCAP, it’s not possible to retrieve just the active UPP from a User’s XML file, instead the whole XML file has to be retrieved. The same code based on the Xerces Java Parser library used in the UPP Interface to handle this same situation is then used to build a DOM tree representation of the XML file, and to extract the active UPP and the preferences it contains.
5. Lastly this information is sent to the Conversation Control and forwarded to the Policy Decision Point.

5.4 Demonstration Scenario

In order to show the potential of UPPs and to demonstrate the implementation the author of this Thesis made, a Use Case was designed detailing an example of how preferences can influence user to user interaction. This Use Case was closely planned with the other members of the NEC SDP team, because only they knew how preferences could be incorporated to the Policies that control the communication session establishment and maintenance in the Policy Decision Point. Also the NEC SDP project is still under development so only a few preferences could actually be implemented as it was, e.g. nor voice mail nor an adress book was implemented, and even for the ones that could be, work was still required on the pre-existing PCC modules Communication Control and Policy Decision Point. So, the preferences that were in fact implemented in the prototype of this Thesis are the following:

- `<only-from-certain-contacts>`: Lists the contact addresses of Users that are allowed to start communication sessions with the User in question; and
- `<session-invitation-types-accepted>`: Lists whether sound and video communication sessions invitations are accepted.

This Use Case was filmed so it can be presented during this Thesis presentation.

5.4.1 Use Case

Short Description

Call sessions are accepted or rejected based on the UPP tags `<only-from-certain-contacts>`, containing the list of users that can establish sessions with the current User (mentioned from now on as white list), and `<session-invitation-types-accepted>`, containing which multimedia session types are accepted (e.g. video and audio).

Actors

- Alice;
- Bob;
- Xaver; and
- Phil.

Pre-conditions

Alice is registered with one user device. The UPP for this device is adequately loaded from the XDMS to the PCC and has the following values:
5. Design and Implementation

- `<only-from-certain-contacts>bob@open-ims.test</only-from-certain-contacts>`
- `<session-invitation-types-accepted text="true" sound="true" video="false" />`

Xaver is registered with another user device. The active UPP for his user device is also adequately loaded and has the following values:

- `<only-from-certain-contacts>phil@open-ims.test</only-from-certain-contacts>`
- `<session-invitation-types-accepted text="true" sound="true" video="true" />`

Bob and Phill each have one user device. The active UPPs for them however are unimportant since these Users will be initiating communication sessions, not receiving them, so the active UPP makes no difference.

All user devices involved are capable of establishing video and sound communication sessions.

**Normal Flow**

1. Bob invites Alice to an audio session.
2. The session is established because Alice’s UPP allows audio sessions and has Bob in its white list.
3. Bob invites Alice to an audio and video session.
4. The invitation is rejected because Alice’s UPP does not allow video sessions.
5. Bob invites Xaver to an audio session.
6. The invitation is rejected because Bob is not in Xaver’s UPP white list.
7. Phil invites Xaver to an audio and video session.
8. The session is established because Xaver’s UPP allows audio and video sessions and has Phil in its white list.

**Alternative Flow**

None.

**Operational and Quality of Experience Requirements**

Alice and Bob both experience the communications in real time. There are no significant delays to any part of the service.
Conclusions and Future Work

Contents

6.1 Conclusion ........................................... 58
6.2 Future Work .......................................... 59
6. Conclusions and Future Work

This chapter finalizes this report by presenting a summary of the design and implementation work done, how it contributes to the standardization process of a UPP for IMS in OMA and elaborates on the business and user value added by this Thesis to the NEC SDP project. Finally, possible avenues of future work are debated.

6.1 Conclusion

This Thesis presents a solution for the need in the mobile industry to personalize the user experience of services. Humans are individually complex and can never fit into a single market segment. They need to be able to make their own choice of how to interact with the services presented to them. IMS allows multimedia services to be created and deployed for devices across fixed and mobile access networks. However, for their full potential to be enabled to the Users, IMS needs to customize on a user level such a diverse offer of services.

To address this issue a UPP was designed in agreement with all requirements by the OMA for this topic. The UPP itself includes all suggested preferences by the OMA but also others of this Thesis author’s design. It was also decided to join the UPP together with the SUP standard, by the same consortium, in the same per user file for extensibility purposes; because despite addressing a different motivation, the SUP uses the same storage and communication protocols as the ones proposed for UPP.

All proposals by the OMA for the supporting architecture were also implemented, which includes the deployment of an XDMS as a network-based storage server for the profiles (an open source XDMS was deployed for this), and XCAP as the communication protocol between such server and both the developed user and service provider XDMCs. A basic Web Interface was created as well, as an end-user interface and to test the developed framework. Finally, a SLEE SBB module was developed to enable a full integration of this Thesis work in the NEC SDP project.

The team colleagues were able to incorporate two preferences in the Policy Decision Point module of the NEC SDP project so that the relevant Policies, and thus the whole IMS system, can reflect them appropriately. This enabled a working prototype of this Thesis. The first preference lists which contact addresses are allowed to initiate communication sessions with the User. The second says if sound and video communication sessions invitations are accepted. A Use Case was designed with these preferences to demonstrate how a UPP improves user to user communications and exemplify the use of UPP in an IMS network. A video of this Use Case was also made to be showed during this Thesis presentation.

Both the extensibility and usability of this Thesis are ensured by the creation of UPPs in XML format. This allows extensions to be made to the XML Schema and the UPP XML files it defines. This is possible both through adding individual attributes or elements (e.g., to address just one new preference) but also through whole new XML namespaces. This way the XDMS would not
suffer any change in the event of an extension to the XML files it stores. Also if new preferences were to be implemented in the NEC SDP project it would only be necessary to add the required new functionalities to the pre-existing PCC modules. The UPP Manager would remain the same.

Implementing the requirements and suggestions of the only consortium, OMA where personalization of services in IMS is being proposed for future services, also ensures the maximum interoperability of the solution built and fits this Thesis requirement of staying as close as possible to models under standardization process.

The fact that this UPP model is still in the initial stages of standardization proves that this level of personalization of services in the IMS is a very recent state-of-the-art topic. For this reason no alternative or competing systems exist as far as is publicly known, although it is likely that at least the OMA members involved in the standardization process (Samsung Electronics, Motorola and the France Télécom group, Orange) are developing their own prototypes and proposals, in the same way this Thesis does for NEC. But the bottom line is that there is no competition to compare this work with.

However, more than the technical details of the design and implementation made in this Thesis, its real value lies in adding user preferences to the NEC SDP User-centric concept of evaluating service provider policy rules, device capabilities and user context, to customize the services offered to the User. This Thesis creates business value because higher valued, and in consequence revenue, services are enabled. User value is also reinforced and augmented because the relation between the User and the services available to him can now be truly personalized.

The bottom line is that this Thesis constitutes a good first step in personalizing the user experience in IMS and opens up new avenues for further work in this topic. In fact, NEC presented a paper to the IEEE Wireless Communications and Networking Conference of 2009 [54], based on the work/subject presented in this Thesis together with the Master Thesis work developed by Moise Ndala, a student colleague at the same team at NEC.

In addition to the skills acquired during this Master Thesis, it has been a great experience working at NEC Laboratories Europe in Heidelberg, Germany.

### 6.2 Future Work

The work presented in this Thesis is at quite an advanced state. There is already a working prototype capable of demonstrating the UPP concept with two preferences successfully implemented. Nevertheless, there is still room for improvement in the prototype itself and new avenues to be explored in the use of a UPP. Figure 6.1 depicts a suggested future architecture for a continuation of this Thesis.

Some aspects which may be worth considering in future work on this subject are:

- To implement more options for the User in the UPP Interface, e.g. create the User Profiles
by editing the Default User Profiles, edit the SUP, edit the preferences for a UPP other than just the two preferences that are implemented in the PCC, etc;

- To integrate the UPP Interface in the NEC SDP user device software installed in the mobile phones, instead of as a Web Page to be accessed via a Web Browser in a desktop pc as it is now;

- To have a SIP connection from the UPP Interface to the Session Management (connection 1 in Figure 6.1). If the previous bullet is implemented, it becomes much easier to implement this SIP connection because the NEC SDP user device software already has such connection implemented, e.g. for session establishment negotiation. This SIP connection would be helpful because if the User changes the active UPP of a user device in the UPP Interface, there is currently no way of signaling this change to the Policy Decision Point, so it can get the new preferences and have the Policies react accordingly, i.e. it would be possible to change the active UPP of a user device during runtime. SIP would be the best protocol for this because it's already implemented between the devices and the Session Management
and the SIP method MESSAGE [SIP] is perfect to carry the information of which is the new active [UPP] and for which device.

This SIP connection could also replace the connection between the UPP Interface and the Data Layer (connection 2 in Figure 5.4) that is currently used to store the active [UPP]s, and to which user devices they apply, in the PCC. The same SIP MESSAGE method could be used. This replacement would increase response time of the system because it would enable such data to be stored in the subscriber-related data server [HSS] which is quicker to be read from than the Data Layer.

- Implement a SIP subscription/notification mechanism between the XDMS and the UPP Manager SBB (connection 2 in Figure 6.1) so the PCC can make subscriptions to XML documents managed by the XDMS and be notified when these change.

For this to be implemented either a SIP Event Subscription Control has to be aggregated to, or built, in the OpenXCAP, or better even to replace OpenXCAP with a SIP capable XDMS like for instance Mobicents XDMS. Since the Mobicents XDMS is built in the Mobicents framework as an SBB it would not even be necessary to use a SIP RA to interact via SIP, since the XDMS would no longer be an external resource.

- Implement more user preferences in the PCC. Only two preferences were implemented but for services to be truly personalized many more have to be implemented.

There is one more avenue of approach containing a big potential for personalization of services in IMS. As suggested by OMA CPM, the User must be able to define the preferences in a UPP based on a per device basis, per address basis or per context basis. The default UPPs suggested by the author of this Thesis in 5.2 reflect these situations, but a new level of personalization could be reached if the active UPPs could be set to change automatically, based on instructions preset by the User or the service provider.

This could be done for instance based on the hour of the day, e.g. program the active UPP of the User's mobile phone to be changed to Driving [UPP] at 8am, Work [UPP] at 9am, Driving [UPP] at 6pm, Home [UPP] at 7pm and Sleeping [UPP] at 23am. Also and more interestingly a mechanism could be set in which the change of supporting network (e.g. WLAN, 3G, 2.5G, etc) triggers the activation of an appropriate UPP e.g. detection of the User’s home WLAN activates the Home [UPP] in the mobile phone, entering an area with a low capacity wireless network like 2G activates a [UPP] which does not allow any video session, or going abroad and accessing the service provider via roaming triggers in the Roaming [UPP]. In fact, the NEC SDP team developed in parallel with this Thesis a context awareness prototype that works with the context simulator Siafu [SIP] to trigger changes directly in the Policies, but could be made to trigger new active UPPs instead.
6. Conclusions and Future Work
Bibliography


Bibliography


[40] Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation, November 2008.


[52] cURL, curl.haxx.se (last accessed October 2009).


Configuration file for OpenXCAP
A. Configuration file for OpenXCAP

[Server]

: IP address and port to listen for requests
: 0.0.0.0 means any address of this host
: address = 0.0.0.0
: port = 8000
: The XCAP Root URI; must not contain any port number. If it has the 'https' scheme,
: the server will listen for requests in TLS mode.
: root = http://xcap.example.com
: The backend to be used for storage and authentication. Current supported
: values are Database and OpenSER.
: backend = Database
: Validate XCAP documents against XML schemas, default is Yes
: document_validiation = Yes

[Authentication]

: The HTTP authentication type, this can be either 'basic' or 'digest'.
: If you're using TLS, it's better to choose 'basic' because the data is encrypted anyway.
: type = basic
: Specify if the passwords are stored as plain text – Yes
: or in a hashed format MD5('username:domain:password') – No
: cleartext_passwords = Yes
: The default authentication realm
: default_realm = localhost
: A list of trusted peers from where XCAP requests are accepted without HTTP authentication
: e.g., trusted_peers = 10.0.0.0/24, 192.168.0.1
: trusted_peers = 127.0.0.1

[TLS]

: Location of X509 certificate and private key that identify this server. The path is
: relative to /etc/openxcap, or it can be given as an absolute path.
: Server X509 certificate
: certificate = tls/server.crt
: Server X509 private key
: private_key = tls/server.key

[Database]

: Configuration settings for Database backend, active if this backend was selected
: in the [Server] section
: The database connection URI for the database with subscriber accounts
: db_url = mysql://openser:openserrw@localhost/openser
: The database connection URI for the database that stores the XCAP documents
: storage_db_url = mysql://openser:openserrw@localhost/openser
: Authentication and storage tables
: subscriber_table = subscriber
: xcap_table = xcap

[OpenSER]

: Configuration settings for OpenSER backend, active if this backend was selected
: in the [Server] section
: The database connection URI for the database with subscriber accounts
: db_url = mysql://openser:openserrw@localhost/openser
: The database connection URI for the database that stores the XCAP documents
: storage_db_url = mysql://openser:openserrw@localhost/openser
: Authentication and storage tables
: subscriber_table = subscriber
: xcap_table = xcap
: The address and port where the XMLRPC Management Interface of OpenSER is listening
: xmlrpc_url = http://localhost:8080
Default User Preferences Profiles
## B. Default User Preferences Profiles

Table B.1: Home UPP

<table>
<thead>
<tr>
<th>Home</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Loud</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to the “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from home</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent, and only from “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, just the headers and with all devices listed as personal</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as personal</td>
</tr>
</tbody>
</table>
### Table B.2: Working UPP

<table>
<thead>
<tr>
<th>Work</th>
<th>Quiet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence info</td>
<td>Available only to the “work” contact group</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from work</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “work” contact group</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text and Sound</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, just the headers and with all devices listed as from work</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as from work</td>
</tr>
</tbody>
</table>

### Table B.3: Meeting UPP

<table>
<thead>
<tr>
<th>Meeting</th>
<th>Quiet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence info</td>
<td>Hide from all</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from home</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Defer delivery of message</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, just the headers and with all devices listed as from work</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as from work</td>
</tr>
</tbody>
</table>
### B. Default User Preferences Profiles

Table B.4: Driving UPP

<table>
<thead>
<tr>
<th>Feature</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Loud</td>
</tr>
<tr>
<td>Presence info</td>
<td>Hide from all</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>False</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>All calls</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Sound</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Yes, just the headers and with all devices Address Book: Yes and with all devices</td>
</tr>
</tbody>
</table>

Table B.5: Sleeping UPP

<table>
<thead>
<tr>
<th>Feature</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Quiet</td>
</tr>
<tr>
<td>Presence info</td>
<td>Hide from all</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from work</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>None</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Defer delivery of message</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: No           Address Book: No</td>
</tr>
</tbody>
</table>
### Table B.6: Roaming UPP

<table>
<thead>
<tr>
<th><strong>Roaming</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Normal</td>
</tr>
<tr>
<td>Presence info</td>
<td>Hide from all</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>No</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>All calls</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: No ((costs would probably be too high to keep updating message status))</td>
</tr>
<tr>
<td></td>
<td>Address Book: No</td>
</tr>
</tbody>
</table>

### Table B.7: Public Places UPP

<table>
<thead>
<tr>
<th><strong>Public Places</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Quiet</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available to all</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>No</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>All calls</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, just the headers and with all devices</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices</td>
</tr>
</tbody>
</table>
### B. Default User Preferences Profiles

#### Table B.8: Personal Mobile UPP

<table>
<thead>
<tr>
<th>Personal Mobile</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Loud</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to: “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from home</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, just the headers and with all devices listed as personal</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as personal</td>
</tr>
</tbody>
</table>

#### Table B.9: Personal Desktop UPP

<table>
<thead>
<tr>
<th>Personal Desktop</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Loud</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to: “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>No</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>No</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, the entire message and with all devices listed as personal</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as personal</td>
</tr>
</tbody>
</table>
Table B.10: Personal Labtop UPP

<table>
<thead>
<tr>
<th>Personal Labtop</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Quiet</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to: “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from home</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “family” and “friends” contact groups</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, the entire message and with all devices listed as personal</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as personal</td>
</tr>
</tbody>
</table>

Table B.11: Work Mobile UPP

<table>
<thead>
<tr>
<th>Work Mobile</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Quiet</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to: “work” contact group</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>Yes</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from work</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “work” contact group</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text and Sound</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, just the headers and with all devices listed as from work</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as from work</td>
</tr>
</tbody>
</table>
### B. Default User Preferences Profiles

Table B.12: Work Desktop UPP

<table>
<thead>
<tr>
<th>Work Desktop</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Quiet</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to: “work” contact group</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>No</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>No</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “work” contact group</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, the entire message and with all devices listed as from work</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as from work</td>
</tr>
</tbody>
</table>

Table B.13: Work Labtop UPP

<table>
<thead>
<tr>
<th>Work Labtop</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ringing Loudness</td>
<td>Quiet</td>
</tr>
<tr>
<td>Presence info</td>
<td>Available only to: “work” contact group</td>
</tr>
<tr>
<td>Sessions while Roaming</td>
<td>No</td>
</tr>
<tr>
<td>3G-WLAN Auto Change</td>
<td>Yes, to the WLAN from work</td>
</tr>
<tr>
<td>Session Invitation acceptance</td>
<td>Only urgent calls, and only from the “work” contact group</td>
</tr>
<tr>
<td>Session Invitation types accepted</td>
<td>Text, Sound and Video</td>
</tr>
<tr>
<td>Message delivery method</td>
<td>For Text, Sound and Video: Store in network based storage</td>
</tr>
<tr>
<td>When to activate session voice mail</td>
<td>All 3 options</td>
</tr>
<tr>
<td>Synchronize from network-based storage</td>
<td>Message: Yes, the entire message and with all devices listed as from work</td>
</tr>
<tr>
<td></td>
<td>Address Book: Yes and with all devices listed as from work</td>
</tr>
</tbody>
</table>
User Profiles XML Schema
C. User Profiles XML Schema
C. User Profiles XML Schema

```xml
<xs:anyAttribute processContents="lax" />  
<xs:extension>  
  <xs:simpleContent>  
    <xs:extension base="xs:string">  
      <xs:attribute ref="xml:lang" />  
    </xs:extension>  
  </xs:simpleContent>  
</xs:extension>  
<xs:complexType>  
  <xs:sequence>  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="hobby">  
      <xs:complexType>  
        <xs:simpleContent>  
          <xs:extension base="xs:string">  
            <xs:attribute ref="xml:lang" />  
          </xs:extension>  
        </xs:simpleContent>  
      </xs:complexType>  
    </xs:element>  
    <xs:element minOccurs="0" maxOccurs="unbounded" namespace="##other" processContents="lax" />  
  </xs:sequence>  
</xs:complexType>  
<xs:complexContent name="hobbiesType">  
  <xs:restriction base="xs:sequence">  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="hobby">  
      <xs:complexType>  
        <xs:simpleContent>  
          <xs:extension base="xs:string">  
            <xs:attribute ref="xml:lang" />  
          </xs:extension>  
        </xs:simpleContent>  
      </xs:complexType>  
    </xs:element>  
  </xs:restriction>  
</xs:complexContent>  
<xs:complexType name="deviceType">  
  <xs:sequence>  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="device">  
      <xs:complexType>  
        <xs:simpleContent>  
          <xs:extension base="xs:string">  
            <xs:attribute ref="xml:lang" />  
          </xs:extension>  
        </xs:simpleContent>  
      </xs:complexType>  
    </xs:element>  
    <xs:element minOccurs="0" maxOccurs="unbounded" namespace="##other" processContents="lax" />  
  </xs:sequence>  
</xs:complexType>  
<xs:complexContent name="devicesNameType">  
  <xs:restriction base="xs:sequence">  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="device">  
      <xs:complexType>  
        <xs:simpleContent>  
          <xs:extension base="xs:string">  
            <xs:attribute ref="xml:lang" />  
          </xs:extension>  
        </xs:simpleContent>  
      </xs:complexType>  
    </xs:element>  
  </xs:restriction>  
</xs:complexContent>  
<xs:complexType name="communicationAddressesType">  
  <xs:sequence>  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="comm-addr">  
      <xs:complexType>  
        <xs:annotation>  
          <xs:documentation>find a way to limit these to the 4 kinds listed in the specs: sip uri, tel uri, e.164 number, or email address</xs:documentation>  
        </xs:annotation>  
        <xs:simpleContent>  
          <xs:extension base="xs:string">  
            <xs:attribute ref="xml:lang" />  
          </xs:extension>  
        </xs:simpleContent>  
      </xs:complexType>  
    </xs:element>  
    <xs:element minOccurs="0" maxOccurs="unbounded" namespace="##other" processContents="lax" />  
  </xs:sequence>  
</xs:complexType>  
<xs:complexContent name="nameType">  
  <xs:restriction base="xs:sequence">  
    <xs:element name="given-name" type="xs:string" />  
    <xs:element name="family-name" type="xs:string" />  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="middle-name" type="xs:string" />  
    <xs:element minOccurs="0" name="name-suffix" type="xs:string" />  
    <xs:element minOccurs="0" maxOccurs="unbounded" name="prefix" type="xs:string" />  
    <xs:element minOccurs="0" maxOccurs="unbounded" namespace="##other" processContents="lax" />  
  </xs:sequence>  
</xs:complexContent>  
<xs:complexType name="addressType">  
  <xs:sequence>  
    <xs:element name="country" type="xs:string" />  
  </xs:sequence>  
</xs:complexType>  
</xs:schema>  
```
<xs:element name="region" type="xs:string"/>
<xs:element name="locality" type="xs:string"/>
<xs:element minOccurs="0" name="area" type="xs:string"/>
<xs:element name="street-name" type="xs:string"/>
<xs:element name="street-number" type="xs:string"/>
<xs:element name="postal-code" type="xs:string"/>
<xs:any minOccurs="0" maxOccurs="unbounded" namespace="#other" processContents="lax"/>
</xs:sequence>
<xs:attribute ref="xml:lang"/>
<xs:anyAttribute processContents="lax"/>
</xs:complexType>
</xs:simpleType>
<xs:complexType name="genderType">
<xs:simpleContent>
<xs:extension base="xs:token">
<xs:enumeration value="male"/>
<xs:enumeration value="female"/>
</xs:extension>
</xs:simpleType>
<xs:simpleType name="publicity-settingsType">
<xs:extension base="xs:string">
<xs:attribute ref="xml:lang"/>
<xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:simpleType>
<xs:complexType name="user-preferences-profileType">
<xs:sequence>
<xs:element name="ringing-loudness" type="ringing-loudnessType"/>
<xs:element name="presence-info" type="presence-infoType"/>
<xs:element name="sessions-while-roaming" type="sessions-while-roamingType"/>
<xs:element name="session-invitation-acceptance" type="session-invitation-acceptanceType"/>
<xs:element name="session-invitation-types-accepted" type="session-invitation-types-acceptedType"/>
<xs:element name="message-delivery-method" type="message-delivery-methodType"/>
<xs:element name="when-to-activate-session-voice-mail" type="when-to-activate-session-voice-mailType"/>
<xs:element name="synchronize-from-network-based-storage" type="synchronize-from-network-based-storageType"/>
<xs:any maxOccurs="unbounded" namespace="#other" processContents="lax"/>
</xs:sequence>
<xs:attribute name="profile-name" type="xs:string" use="required"/>
<xs:attribute name="wlan-auto-change" type="xs:boolean" use="required"/>
<xs:attribute processContents="lax"/>
</xs:complexType>
</xs:simpleType>
<xs:complexType name="ringing-loudnessType">
<xs:restriction base="xs:token">
<xs:enumeration value="loud"/>
<xs:enumeration value="normal"/>
<xs:enumeration value="quiet"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="presence-infoType">
<xs:extension base="xs:string">
<xs:attribute ref="xml:lang"/>
<xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:simpleType>
<xs:complexType name="session-invitation-acceptanceType">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="available-only-to">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute ref="xml:lang"/>
<xs:anyAttribute processContents="lax"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="hide-from-all" type="xs:boolean" use="required"/>
<xs:attribute name="available-to-all" type="xs:boolean" use="required"/>
</xs:complexType>
</xs:simpleType>
<xs:complexType name="sessions-while-roamingType">
<xs:element name="available"/>
<xs:complexType>
<xs:element name="video" type="xs:boolean" use="required"/>
<xs:element name="sound" type="xs:boolean" use="required"/>
<xs:element name="text" type="xs:boolean" use="required"/>
</xs:complexType>
</xs:element>
</xs:complexType>
<xs:element name="unavailable"/>
C. User Profiles XML Schema

```xml
<xs:complexType name="session-invitation-acceptanceType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="only-from-certain-contacts">
      <xs:complexType>
        <xs:complexContent>
          <xs:restriction base="xs:string">
            <xs:enumeration value="all-calls" type="xs:boolean" use="required"/>
            <xs:enumeration value="only-urgent-calls" type="xs:boolean" use="required"/>
            <xs:enumeration value="only-private-calls" type="xs:boolean" use="required"/>
            <xs:enumeration value="no-one" type="xs:boolean" use="required"/>
          </xs:restriction>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="video" type="xs:boolean" use="required"/>
    <xs:element name="sound" type="xs:boolean" use="required"/>
    <xs:element name="text" type="xs:boolean" use="required"/>
    <xs:element name="message" type="message-delivery-methodType"/>
    <xs:element name="network" type="network-storageType"/>
    <xs:element name="contact" type="phone-contactType"/>
    <xs:element name="storage" type="storageType"/>
  </xs:sequence>
</xs:complexType>
```

```xml
<x:complexType name="session-invitation-types-acceptedType">
  <xs:sequence>
    <xs:element name="in-contacts" type="in-contacts"/>
    <xs:element name="in-invitations" type="in-invitations"/>
    <xs:element name="in-chats" type="in-chats"/>
  </xs:sequence>
</xs:complexType>
```

82
<xs:simpleContent>
  <xs:extension base="xs:string">
    <xs:attribute ref="xml:lang"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element name="address-book">
</xs:complexType>
</xs:complexType>
</xs:element name="check">
</xs:complexType>
</xs:complexType>
</xs:element maxOccurs="unbounded" name="with-which-devices">
</xs:complexType>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:element name="uncheck"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:element>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element name="uncheck"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:schema>