Adaptive Personal Mobile Communication

Service Architecture and Protocols

Theo G. Kanter

Laboratory of Communication Networks
Department of Microelectronics and Information Technology
Royal Institute of Technology
Stockholm, Sweden
Adaptive Personal Mobile Communication

Service Architecture and Protocols

Theo G. Kanter

A dissertation submitted to the Royal Institute of Technology in partial fulfillment of the requirements for the degree of Doctor of Technology.

November 2001

TRITA-IT AVH 01:01
ISSN 1403-5286
ISRN KTH/IT/AVH--01/01

Laboratory of Communication Networks
Department of Microelectronics and Information Technology
Royal Institute of Technology
Stockholm, Sweden
“It matters if you just don’t give up.”

— Stephen Hawking
Acknowledgments

Prof. Gerald Q. Maguire Jr. has been an invaluable source of inspiration ever since we met in early 1996, when I was pioneering the first ‘Voice over IP’ services. I would like to express my deep gratitude to my advisor Professor Maguire, who’s foresight, encouragement, enthusiasm, and inspiration set an example for my research, and his tireless support and advice helped me to start asking the real questions, and in successive projects produce step-by-step proof of the feasibility of mobile multimedia on wireless Internet and its potential for user-centric communication. I was also very fortunate to have Dr. Mark T. Smith of HP-labs as my second advisor for my research. Mark generously shared his views and advice concerning how to look upon services from a consumer perspective, which I then could contrast to the telecom mindset in the ‘mobile valley’ of north Stockholm. I also thank the late Tech.Lic. Anders Carlsson - the CEO of Ellemtel until his early death in 1997, whose leadership inspired us to find new ways of communication - for preparing the ground for this research. I would also like to recognize the support of my management, Aleksander Marlevi (in 1998 the vice-CEO of Ellemtel where this thesis work began), Håkan Eriksson (head of Ericsson Research, where this thesis work was completed in 2001), and in particular Rolf Leidhammar (director of Applications & Software research) for demanding clear explanations about the direction and results of this research, thus helping me to be clearer about its purpose, and further clearing the path for the dissemination of results at Ericsson. This research investigated the impact of Internet on many related components and would probably neither have been undertaken or completed without the cooperation and support of many colleagues in different laboratories, among whom I would like to extend special thanks to Claes Frisk, Henrik Gustafsson, Tom Rindborg, Dan Sahlin, Anders Danne, Alberto Escudero-Pascual, and Joe Mitola III for their generous help, enlightening and valuable suggestions, stimulating discussions, support, and cooperation. I would also like to thank Peter Peldan, Rabbe Fogelholm, Eric Nissen, Carl-Wilhelm Welin, Bjarne Däcker, Hans Nilsson, Håkan Larsson, several ex-jobbers: Jing Cheng, Hilde Hassli, & Max Blommé who joined the application architecture lab, Erik Wiklander & Sofia Gunnarsson, and the students who participated in the HotTown project (Vahid Mosavat, Niclas Becker, Pontus Svärd, Per Hallin, and Rickard Reite) for their help along my path to this dissertation. Joe Mitola’s work concerning Cognitive Radio has been an additional source of inspiration through his focus on mobiles being aware of their communication context, while my work has focused more on networking and service architecture aspects of mobile applications. This research was conducted with partial support from the Swedish Foundation for Strategic Research's Personal Computing and Communication (PCC) research program. Dr. Bengt-Arne Molin's tireless efforts as (former) program director to establish me as an industrial doctoral student should be mentioned. Within PCC, Professors Carl-Gustaf Jansson, Björn Pehrson, and Jens Zander provided an important research context for my work.

This dissertation is dedicated to my wife Alice and three daughters - Anna, Sofia, and Beatrice - for their unending support, love and patience, during the three and a half years it took to complete this.

Abstract

This dissertation addresses the applications, which are enabled by combining results from ubiquitous computing, mobile networks, and Voice over IP (VoIP) technologies and support for enabling users and applications to adapt to a more diverse communication environment. The cost to transmit digital information end-to-end is dropping dramatically, along with a tremendous increase in the available bandwidth in access networks for both fixed and wireless access. These trends have been accelerated by the large-scale deployment of broadband networks, which are an ideal base for adding wireless LAN extensions, providing order of magnitude increases in wireless bandwidth in comparison to the projected third-generation wireless networks. This motivates their complementary use in order to meet new user demand, and quickens the pace of on-going deregulation and separation of roles regarding provisioning of network access, services, and transport. Furthermore, short-range radio-link technologies facilitate new ways of interaction both between people & devices and between devices. The price/performance of end-user electronics drops along with a tremendous increase in computational power; this increased processing power can be used to deal with the increasingly diverse wireless infrastructure in an optimal way. Furthermore, these developments which have created affordable communication between users, computational devices, and resources have also removed a number of the limitations on the kind of services that were previously possible. We are now able to build new classes of end-user applications solely on top of IP (in particular where wireless access is involved). While a number of requirements on the network have been relaxed, this raises new questions, not only about which applications are enabled, but also about how users, mobile artifacts, and virtual objects can negotiate for services with a minimum of a-priori, shared knowledge, which also enables these entities to adapt to a diverse wireless communication infrastructure and available resources. These new requirements that are placed on the infrastructure, call for completely rethinking established service architectures for public mobile networks. This rethinking is expected to have far-reaching implications on how actors in the converging computing and communications industries will deliver services, and what services they will deliver.

The dissertation first examines the feasibility of delivering mobile multimedia over wireless links with end-to-end IP connectivity. This is followed by an analysis of different service architectures for delivering these services and the necessary properties of a open model for describing, managing, using, and exchanging service components. A novel approach, called a ‘Mobile Interactive Space’, is presented that provides interaction between representations of people, devices, and resources, connected by ubiquitous communication. The research leading to this doctorate has created new network and system level models for building applications in which the new requirements and design-rules can be mapped, along with a synthesized protocol and specification language for describing, managing, using, and exchanging service components in a ‘Mobile Interactive Space’. An analysis of the implications of the above approach for new business models is presented, along with results from experiments giving evidence of the feasibility of the design of the architecture and its components, and for building solutions that deliver the new services and enable users to cope with a heterogeneous and deregulated communication environment. Finally, I discuss some topics for continued work regarding questions that were unearthed by the dissertation or for which no conclusive answer could be given.
Keywords: Mobile Computing, Personal Communication, Mobile Agents, Wireless Networks.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>i</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Telephony Services ........................................................................................... 1
   1.1.1 Breaking out ....................................................................................... 1
   1.1.2 A Voice is heard on the Net ................................................................ 2
   1.1.3 Convergence is fueled by VoIP .......................................................... 3
1.2 Wireless Industry at the Crossroads ................................................................. 3
   1.2.1 3G and Beyond................................................................................... 3
   1.2.2 Introduction to “πG” ................................................................. 3
   1.2.3 Convergence? ..................................................................................... 4
   1.2.4 Towards a (Multimedia-Enabled) Wireless Internet .......................... 5
1.3 Motivations: Towards Ubiquitous Computing & Communication .................. 6
   1.3.1 Advances in Mobile Computing ........................................................ 6
   1.3.2 Scenarios, New Models Of Communication ............................................. 6
   1.3.3 Specification and Negotiation of Service Behavior ........................... 8
   1.3.4 The Hypothesis and Approach ................................................................ 8
   1.3.5 Revisiting the Role of VoIP ........................................................... 9
1.4 Specific Problem Statement ........................................................................... 10
1.5 Overview ........................................................................................................ 11
1.6 Contributions and References ........................................................................ 12
1.7 Summary of my published papers and other documents ................................. 14

## 2 Background

2.1 Fixed Access to Internet ......................................................................................... 19
   2.1.1 Dialup Access .................................................................................... 19
   2.1.2 xDSL ................................................................................................ 20
   2.1.3 ATM ................................................................................................. 20
   2.1.4 Dynamic synchronous Transfer Mode (DTM) .............................................. 20
   2.1.5 Packet-oriented (Ethernet/IP) ........................................................... 20
   2.1.6 Cable Networks ................................................................................ 20
2.2 Wireless Access to Internet .................................................................................. 21
   2.2.1 Satellites ........................................................................................... 21
   2.2.2 Local Area Wireless Communication .................................................. 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.3 Short-range Wireless Access (Bluetooth)</td>
<td>22</td>
</tr>
<tr>
<td>2.2.4 Cellular networks</td>
<td>22</td>
</tr>
<tr>
<td>2.3 Wireless QoS</td>
<td>22</td>
</tr>
<tr>
<td>2.3.1 Bandwidth</td>
<td>22</td>
</tr>
<tr>
<td>2.3.2 Resource Reservation</td>
<td>23</td>
</tr>
<tr>
<td>2.3.3 Latency</td>
<td>23</td>
</tr>
<tr>
<td>2.3.4 Robustness</td>
<td>23</td>
</tr>
<tr>
<td>2.3.5 Encoding, Decoding, and Transcoding</td>
<td>23</td>
</tr>
<tr>
<td>2.3.6 Header Compression</td>
<td>24</td>
</tr>
<tr>
<td>2.3.7 Summary</td>
<td>24</td>
</tr>
<tr>
<td>2.4 (Mobile) Devices</td>
<td>24</td>
</tr>
<tr>
<td>2.4.1 Enhanced Capabilities</td>
<td>24</td>
</tr>
<tr>
<td>2.4.2 Utilizing Wireless Access</td>
<td>24</td>
</tr>
<tr>
<td>2.4.3 Mobile Users</td>
<td>24</td>
</tr>
<tr>
<td>2.5 Changes in the Communication Industry</td>
<td>25</td>
</tr>
<tr>
<td>2.5.1 Introduction</td>
<td>25</td>
</tr>
<tr>
<td>2.5.2 Deployment of Services</td>
<td>26</td>
</tr>
<tr>
<td>2.5.3 Telecom Model</td>
<td>26</td>
</tr>
<tr>
<td>2.5.4 Datacom Model</td>
<td>26</td>
</tr>
<tr>
<td>2.5.5 Cost-effectiveness</td>
<td>26</td>
</tr>
<tr>
<td>2.5.6 Consumer Model of Computing Capable End-User Devices</td>
<td>26</td>
</tr>
<tr>
<td>2.6 Business Model Implications</td>
<td>27</td>
</tr>
<tr>
<td>2.6.1 Packaging the service with the network</td>
<td>27</td>
</tr>
<tr>
<td>2.6.2 3d Party, Consumer-model</td>
<td>27</td>
</tr>
<tr>
<td>2.6.3 Infrastructure Development Phases</td>
<td>28</td>
</tr>
<tr>
<td>2.6.4 Cellular Pitfalls</td>
<td>28</td>
</tr>
<tr>
<td>2.6.5 Wireless Wild-West</td>
<td>28</td>
</tr>
<tr>
<td>2.7 Chapter summary and conclusions</td>
<td>29</td>
</tr>
<tr>
<td>3 Related Work</td>
<td>31</td>
</tr>
<tr>
<td>3.1 Limitations in Existing Telecom Service Architectures</td>
<td>31</td>
</tr>
<tr>
<td>3.2 Computer Telephony Integration</td>
<td>31</td>
</tr>
<tr>
<td>3.3 ATM, B-ISDN</td>
<td>32</td>
</tr>
<tr>
<td>3.4 Tunneling Telecom Services over IP- Networks</td>
<td>33</td>
</tr>
<tr>
<td>3.5 Intelligent Networks (IN)</td>
<td>33</td>
</tr>
<tr>
<td>3.6 3G Roadmap to Mobile Internet</td>
<td>33</td>
</tr>
<tr>
<td>3.6.1 2G (Second Generation Cellular)</td>
<td>34</td>
</tr>
<tr>
<td>3.6.2 WAP</td>
<td>34</td>
</tr>
<tr>
<td>3.6.3 2.5G</td>
<td>36</td>
</tr>
<tr>
<td>3.6.4 Parlay</td>
<td>36</td>
</tr>
<tr>
<td>3.6.5 3G Phase 1</td>
<td>37</td>
</tr>
<tr>
<td>3.6.6 Virtual Home Environment (VHE)</td>
<td>37</td>
</tr>
<tr>
<td>3.6.7 3G Phase 2</td>
<td>38</td>
</tr>
<tr>
<td>3.6.8 3G Service Delivery Framework</td>
<td>38</td>
</tr>
<tr>
<td>3.6.9 QoS Support for Mobile Users</td>
<td>39</td>
</tr>
<tr>
<td>3.6.10 Summary 3G Roadmap</td>
<td>39</td>
</tr>
<tr>
<td>3.7 πG</td>
<td>39</td>
</tr>
<tr>
<td>3.7.1 Business Model Independent Service Delivery Framework</td>
<td>40</td>
</tr>
<tr>
<td>3.7.2 Beyond Circuit Switching and Call Models</td>
<td>41</td>
</tr>
</tbody>
</table>
3.8 Internet Multimedia Architectures ................................................................. 41
3.8.1 ITU-T's H.323 .................................................................................. 41
3.8.2 Call Management Agents ................................................................. 42
3.8.3 IETF's Session Initiation Protocol ................................................... 42
3.8.4 IETF's Session Description Protocol ............................................... 43
3.8.5 IETF's Call Processing Language .................................................... 43
3.8.6 JAIN ................................................................................................. 43
3.8.7 SIPlets .............................................................................................. 43
3.9 Open Communication on Internet ............................................................. 43
3.9.1 CORBA ............................................................................................ 44
3.9.2 Service Location Protocol ................................................................ 44
3.9.3 Plug & Play Architectures for Pervasive Computing ...................... 44
3.9.4 Microsoft '.NET' .............................................................................. 45
3.9.5 The Semantic Web (W3C) ................................................................ 46
3.10 Ubiquitous Computing and Communication .......................................... 47
3.10.1 Linda (T-Spaces) .............................................................................. 47
3.10.2 Context- & location aware computing ............................................. 47
3.10.3 Agents Overview .............................................................................. 47
3.10.4 Mobile Agents .................................................................................. 47
3.10.5 Mobile Code ..................................................................................... 48
3.11 Knowledge Representation and Manipulation ........................................ 48
3.11.1 KIF and KQML ................................................................................ 48
3.11.2 Knowledge Maintenance ................................................................. 49
3.11.3 RKRL ............................................................................................... 50
3.12 Feature Interaction ............................................................................... 50
3.13 Chapter summary and conclusions .......................................................... 51

4 A New Service Architecture for the Internet .............................................. 53
4.1 Terminology .......................................................................................... 53
4.2 Enabling Technologies ........................................................................... 53
4.2.1 Voice over IP (VoIP) ........................................................................ 54
4.2.2 SIP .................................................................................................... 54
4.2.3 Mobile Agents .................................................................................. 54
4.3 Guiding Vision: The Mobile Interactive Space ........................................ 54
4.4 Overview of the Service Architecture Framework ................................... 55
4.4.1 Entities modeled as Agents and (Software) Objects ........................ 55
4.4.2 Protocol Stack ................................................................................. 56
4.4.3 Physical and Link Layers ................................................................. 57
4.4.4 Network Layer ................................................................................. 57
4.4.5 Transport- & Session-Layer ............................................................. 57
4.4.6 Application Layer ............................................................................ 59
4.4.7 Extension Layer .............................................................................. 60
4.4.8 Mobile Knowledge Layer ............................................................... 60
4.5 Merits of the Solution ............................................................................ 61
4.5.1 Removing previous limitations while retaining scalability .............. 61
4.5.2 Adaptive Responses to Communication Events .............................. 62
4.5.3 Seamless Services and User Interfaces .......................................... 62
4.5.4 Ad hoc Negotiation of Services ...................................................... 63
4.5.5 Intelligent Responses to a Communication Event ........................... 64
4.5.6 Conclusions regarding the merits of my solution ......................... 64
4.6 The Design of the Architecture ......................................................... 65
4.7 Operational Environment ................................................................. 65
4.7.1 Integration of Legacy Services ....................................................... 65
4.7.2 (Logically) Off-line Behavior ......................................................... 66
4.7.3 Adding Context-Awareness ............................................................ 66
4.7.4 Management of Location Data ....................................................... 67
4.7.5 Human-Computer Interaction issues .............................................. 67
4.7.6 Agent Communication ................................................................. 68
4.7.7 Peer-to-Peer or Server mode ......................................................... 68
4.7.8 Personal Agent ............................................................................. 68
4.8 Agent Properties ............................................................................. 68
4.8.1 Agent Profiles ............................................................................... 70
4.8.2 Types ............................................................................................ 70
4.8.3 Actions .......................................................................................... 71
4.8.4 States ............................................................................................ 71
4.9 Mobility ............................................................................................ 71
4.9.1 Personal/Entity Mobility ............................................................... 72
4.9.2 Service Mobility ............................................................................ 72
4.9.3 Device Mobility ............................................................................. 72
4.9.4 Session Mobility .......................................................................... 73
4.9.5 Roaming ......................................................................................... 73
4.9.6 Handover ....................................................................................... 74
4.9.7 Mobile multimedia handovers ..................................................... 74
4.9.8 Remote Networks .......................................................................... 75
4.10 Invocation of Resources (e.g., VoIP) .................................................... 75
4.10.1 Integration of Devices .................................................................. 76
4.10.2 Example ....................................................................................... 76
4.11 Concluding remarks concerning this new service architecture ............ 76

5 Mobile Service Knowledge (MSK) ...................................................... 79
5.1 Introduction ....................................................................................... 79
5.1.1 Direction ....................................................................................... 80
5.1.2 Structure and Computability .......................................................... 80
5.1.3 Example ....................................................................................... 81
5.2 Specification ..................................................................................... 81
5.3 General Properties ........................................................................... 82
5.3.1 Mobilets ....................................................................................... 82
5.3.2 Relations ..................................................................................... 83
5.3.3 Synthesizing MSK ....................................................................... 84
5.3.4 Context Management ................................................................. 84
5.3.5 Meta-Contexts ............................................................................. 85
5.3.6 Inheritance and Aggregation ....................................................... 85
5.3.7 Ontologies .................................................................................... 85
5.4 External Properties .......................................................................... 86
5.4.1 Caching ....................................................................................... 86
5.4.2 Mobility ....................................................................................... 86
5.4.3 Export/Import ............................................................................. 86
5.4.4 Security ....................................................................................... 86
# Active Context Memory

## 6.1 Overview

## 6.2 Machine Learning

### 6.2.1 Example

### 6.2.2 Requirements

### 6.2.3 Components

## 6.3 Architecture and Design

### 6.3.1 Knowledge Base

### 6.3.2 Knowledge Management

### 6.3.3 Knowledge Extension

### 6.3.4 Knowledge Scoping

### 6.3.5 Concepts Generator

### 6.3.6 Garbage Collector

### 6.3.7 Integrity Checker

### 6.3.8 Evaluator Tool

### 6.3.9 Inference Engine

### 6.3.10 Reconciliation of knowledge structures

### 6.3.11 Foreground / Background Knowledge & Handling

## 6.4 Design Decisions

## 6.5 Mobile Knowledge Repository

## 6.6 Conclusions concerning an Active Context Memory

# An eXtensible Service Protocol

## 7.1 Purpose

## 7.2 Steps in Negotiation

### 7.2.1 State Diagram

### 7.2.2 Service Discovery

### 7.2.3 Event Routing

## 7.3 Object Access

### 7.3.1 Design Rationale

### 7.3.2 Requesting and Initializing Profiles

### 7.3.3 Notifications

### 7.3.4 Extensions of Profiles

### 7.3.5 Extension of Mobile Service Knowledge

## 7.4 Mobility

## 7.5 Mobile Interactive Spaces

### 7.5.1 Mobile Resource Awareness

### 7.5.2 Dissemination Algorithm

### 7.5.3 Access Control

## 7.6 Negotiation
7.6.1 Collective Behavior ................................................................. 111
7.6.2 Monitoring Behavior ............................................................... 112
7.6.3 Exchanging Behavior ............................................................... 112

7.7 Summary and conclusions concerning an Extensible Service Protocol ..... 113

8 Network Aspects and Requirements 115

8.1 Introduction .................................................................................. 115

8.2 Impact on Service Delivery Frameworks ........................................... 115
8.2.1 (Advanced) Intelligent Networks ............................................. 115
8.2.2 CTI ......................................................................................... 116
8.2.3 GSM ....................................................................................... 116
8.2.4 3G IP-MM .............................................................................. 116
8.2.5 Network Operators ................................................................. 117
8.2.6 Application Providers ............................................................... 118
8.2.7 Credential and Access Providers .............................................. 119
8.2.8 Meta-Application Providers .................................................... 119
8.2.9 Examples of Uses .................................................................... 120

8.3 Networking Aspects ........................................................................ 120
8.3.1 Presence and Location Support ................................................. 120
8.3.2 Feature Interaction .................................................................... 122
8.3.3 Network vs. Client Considerations ......................................... 124
8.3.4 Mobile Agents .......................................................................... 124
8.3.5 Synchronization ........................................................................ 124
8.3.6 Dissemination .......................................................................... 125
8.3.7 Service Prediction ...................................................................... 126
8.3.8 Security ................................................................................... 126
8.3.9 Privacy .................................................................................... 127
8.3.10 Preserving Peer-to-Peer Networking (Firewalls, NAT) .......... 127
8.3.11 Trust and Control .................................................................... 127
8.3.12 Power Management ............................................................... 128
8.3.13 Computational Resources ....................................................... 128
8.3.14 Persistence .............................................................................. 128
8.3.15 Scalability .............................................................................. 128
8.3.16 Step-wise Introduction ............................................................ 129
8.3.17 Network Economics ............................................................... 129
8.3.18 Packaging Service Behavior .................................................. 129
8.3.19 Service Repositories and Management ................................... 130
8.3.20 Service Data-mining ............................................................... 131
8.3.21 Manageability ......................................................................... 131

8.4 Chapter Summary and Conclusions .............................................. 131

9 Enabling Mobile Communication 133

9.1 Introduction .................................................................................. 133

9.2 Modes of Human Computer Interaction ........................................ 134
9.2.1 Existing Modes ........................................................................ 134
9.2.2 New Modes ........................................................................... 134

9.3 Service Delivery ........................................................................... 135
9.3.1 Delivering Existing Services .................................................... 135
9.3.2 Delivering New Services................................................................. 135
9.4 Experimental Network and Results.................................................. 136
  9.4.1 Brf. Bågen .............................................................................. 137
  9.4.2 Wireless Access ........................................................................ 137
  9.4.3 Direct Internet Access .............................................................. 138
  9.4.4 SIP naming, localization, and session invocation ......................... 138
  9.4.5 Security .................................................................................. 138
  9.4.6 Mobility ................................................................................. 139
  9.4.7 Handoff Measurement .............................................................. 139
  9.4.8 Quality of Service .................................................................... 138
  9.4.9 Capacity .................................................................................. 140
  9.4.10 Hosting and Interworking ......................................................... 141
  9.4.11 Summary and conclusions based on the experimental network .... 141
9.5 Ad Hoc Internet Access and Services ................................................. 142
  9.5.1 Discovery and Synthesis of MSK ............................................ 142
  9.5.2 Scenario 1 .............................................................................. 147
  9.5.3 Scenario 2 .............................................................................. 148
  9.5.4 Scenario 3 .............................................................................. 149
  9.5.5 Context Stacks ......................................................................... 150
  9.5.6 Other scenarios ...................................................................... 151
  9.5.7 Section Summary and Conclusions........................................... 151
9.6 Virtual Office Prototype ...................................................................... 152
  9.6.1 Application ............................................................................. 153
  9.6.2 Agents .................................................................................... 154
  9.6.3 Scenarios .............................................................................. 154
  9.6.4 Conclusions concerning the Virtual Office Prototype ................. 157
9.7 Multimedia Service Delivery .............................................................. 158
  9.7.1 Software Architecture .............................................................. 159
  9.7.2 Voice over IP (VoIP) over Fixed Access to Internet .................... 159
  9.7.3 VoIP over a Wireless LAN Link (WLAN) .................................. 159
  9.7.4 VoIP over a Wide-Area Wireless Link (GSM, GRPRS) ............... 160
  9.7.5 Asynchronous Media ............................................................... 160
  9.7.6 Mobile-Aware Media Player .................................................... 160
  9.7.7 Hotspot Density ...................................................................... 161
  9.7.8 Near-Isochronous Communication .......................................... 161
  9.7.9 Analysis ................................................................................. 161
  9.7.10 Conclusions concerning multimedia service delivery ............... 163
9.8 Smart Delivery Prototype .................................................................... 163
  9.8.1 Scenarios .............................................................................. 164
  9.8.2 Opportunistic Communication .................................................. 165
  9.8.3 Conclusions concerning a smart delivery prototype ................... 166
9.9 Mixed-Reality Bågen Prototype ......................................................... 167
  9.9.1 Prototype Overview ................................................................. 168
  9.9.2 User Scenario ........................................................................ 169
  9.9.3 Conclusions based on this mixed-reality prototype .................... 170
9.10 Chapter Summary............................................................................... 170

10 Conclusions ..................................................................................... 171
  10.1 Overview .................................................................................... 171
11 Future Work 177
11.1 Unsolved Problems and Remaining Open Issues .............................................. 177
11.2 Near Term ........................................................................................................ 177
   11.2.1 Management of Mobile Service Knowledge ........................................... 177
   11.2.2 Feature Interaction .................................................................................. 178
   11.2.3 Security .................................................................................................... 178
   11.2.4 Smart Media Delivery .............................................................................. 179
   11.2.5 Mobile Device Middleware ...................................................................... 180
   11.2.6 Billing & Charging .................................................................................. 180
11.3 Longer Term .................................................................................................... 181
   11.3.1 Trust & Control and Ambient Intelligence .................................................. 181
   11.3.2 Machine Learning ................................................................................... 181
11.4 Future Prospects ............................................................................................. 182
   11.4.1 Cognitive Internet .................................................................................... 182
   11.4.2 Mobile Brains .......................................................................................... 182
   11.4.3 Teleportation ............................................................................................ 183

References 185

Acronyms and Abbreviations 199
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>PhoneDoubler™ Solution</td>
<td>19</td>
</tr>
<tr>
<td>3-1</td>
<td>The CTI Model</td>
<td>31</td>
</tr>
<tr>
<td>3-2</td>
<td>Mobile Internet with WAP 1.0</td>
<td>33</td>
</tr>
<tr>
<td>3-3</td>
<td>Service Architectures beyond Second Generation Mobile Networks</td>
<td>34</td>
</tr>
<tr>
<td>3-4</td>
<td>Access Networks in 3G and Beyond</td>
<td>35</td>
</tr>
<tr>
<td>3-5</td>
<td>Parlay Framework (left) and Parlay/SIP Prototype (right)</td>
<td>36</td>
</tr>
<tr>
<td>3-6</td>
<td>Virtual Home Environment Realization</td>
<td>38</td>
</tr>
<tr>
<td>3-7</td>
<td>(a) Service Platform in Home Network, (b) External Service Platform</td>
<td>39</td>
</tr>
<tr>
<td>3-8</td>
<td>Business Model Independent Service Delivery Framework</td>
<td>40</td>
</tr>
<tr>
<td>3-9</td>
<td>User-Centric Communication</td>
<td>41</td>
</tr>
<tr>
<td>3-10</td>
<td>The Call Management Agent (CMA) Architecture</td>
<td>42</td>
</tr>
<tr>
<td>3-11</td>
<td>UPnP Protocol Stack</td>
<td>45</td>
</tr>
<tr>
<td>4-1</td>
<td>Mobile Interactive Space</td>
<td>54</td>
</tr>
<tr>
<td>4-2</td>
<td>Service Architecture Framework</td>
<td>56</td>
</tr>
<tr>
<td>4-3</td>
<td>Protocol Stack Overview</td>
<td>57</td>
</tr>
<tr>
<td>4-4</td>
<td>(Mobile) Agents</td>
<td>58</td>
</tr>
<tr>
<td>4-5</td>
<td>Personal (Mobile) Agent</td>
<td>58</td>
</tr>
<tr>
<td>4-6</td>
<td>Communicating Entities</td>
<td>60</td>
</tr>
<tr>
<td>4-7</td>
<td>Mobile Service Knowledge (MSK) atop XSP</td>
<td>61</td>
</tr>
<tr>
<td>4-8</td>
<td>Operational Environment</td>
<td>65</td>
</tr>
<tr>
<td>4-9</td>
<td>Personal Agent Migration</td>
<td>67</td>
</tr>
<tr>
<td>4-10</td>
<td>Personal Agent General Properties</td>
<td>68</td>
</tr>
<tr>
<td>4-11</td>
<td>Mobility support in the Service Architecture</td>
<td>71</td>
</tr>
<tr>
<td>4-12</td>
<td>User Mobility Scenario</td>
<td>73</td>
</tr>
<tr>
<td>4-13</td>
<td>Mobile-IP Extensions</td>
<td>74</td>
</tr>
<tr>
<td>4-14</td>
<td>Example</td>
<td>76</td>
</tr>
<tr>
<td>5-1</td>
<td>Access Relations</td>
<td>81</td>
</tr>
<tr>
<td>5-2</td>
<td>Synthesizing of MSK</td>
<td>84</td>
</tr>
<tr>
<td>5-3</td>
<td>Relations</td>
<td>84</td>
</tr>
<tr>
<td>5-4</td>
<td>A Person</td>
<td>85</td>
</tr>
<tr>
<td>6-1</td>
<td>ACM, XSP Events</td>
<td>91</td>
</tr>
<tr>
<td>6-2</td>
<td>ACM Structure</td>
<td>93</td>
</tr>
<tr>
<td>6-3</td>
<td>Personal Mobile Agent design</td>
<td>94</td>
</tr>
<tr>
<td>6-4</td>
<td>Knowledge Scoping</td>
<td>95</td>
</tr>
<tr>
<td>6-5</td>
<td>Use of Context Stacks</td>
<td>96</td>
</tr>
<tr>
<td>6-6</td>
<td>Resource Contracts</td>
<td>97</td>
</tr>
<tr>
<td>6-7</td>
<td>Reconciling Knowledge Structures</td>
<td>99</td>
</tr>
<tr>
<td>6-8</td>
<td>Meeting</td>
<td>100</td>
</tr>
<tr>
<td>6-9</td>
<td>Merged Structures</td>
<td>100</td>
</tr>
<tr>
<td>7-1</td>
<td>XSP &amp; Events</td>
<td>103</td>
</tr>
<tr>
<td>7-2</td>
<td>Protocol Stack</td>
<td>103</td>
</tr>
<tr>
<td>7-3</td>
<td>XSP State Diagram &amp; Messages</td>
<td>105</td>
</tr>
<tr>
<td>7-4</td>
<td>Merging Ontologies</td>
<td>112</td>
</tr>
<tr>
<td>8-1</td>
<td>Reference Model</td>
<td>115</td>
</tr>
<tr>
<td>8-2</td>
<td>Network Providers</td>
<td>117</td>
</tr>
<tr>
<td>8-3</td>
<td>Meta-Application Providers</td>
<td>119</td>
</tr>
<tr>
<td>8-4</td>
<td>Meeting Meta-Context</td>
<td>120</td>
</tr>
<tr>
<td>8-5</td>
<td>Group Meta-Context</td>
<td>120</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

This chapter presents the background to & motivations for this research, and lists its contributions and publications: it provides an overview of trends in the communications industry and what mechanisms are at work, which motivated this research and cause it to be important. I describe what problems are addressed by the dissertation, and the contribution of the dissertation - detailing how it has solved these problems.

1.1 Telephony Services

Or “why existing telecom networks and service architectures are limited with respect to service growth.”

Today’s world of personal communication has its roots in the switched telephony networks. The convergence of telecom and datacom networks has lead to a wide range of experimentation to create new services and to exploit opportunities to circumvent the limitations due to the restricted user interface of telephones. However, the service that these switched telephony networks were designed for was setting up fixed communication channels for bi-directional speech. Thus, adding service functionality to these networks can only be achieved with a separate entity that monitors events in the communication channels. The events that occur are principally the signaling to set up a call connection and the termination of the call. Telephony services in general, as implemented by these entities, simply speak in terms of call events.

Examples of technologies to implement more advanced services are Computer Telephony Integration (CTI)\(^1\) [5] or so-called Advanced Intelligent Networks (AIN) [6]. Both share the property that the services control only one class of events, which are the origination of a session (i.e., initiating a telephone call). The resources in the user-interface (e.g. data and voice) have to be coordinated by an entity in the network. In the case of AIN, key presses by a user that relate to user services are relayed to dedicated network nodes for further call processing.

1.1.1 Breaking out

Overcoming the limitation of this call event model means moving the locus of control all the way out to the user interfaces, which means out of the network. Thus the media that are included in the user interface have to be delivered by an access mechanism that is transparent, i.e. unaware of the specific service. This means that the points where sources of information are included in the user application and the application itself share a common access to the network. Hence, the network should provide a flat transport medium for digital information, which is routed to the requested

---

1. All acronyms are listed starting on page 153 in “Acronyms and Abbreviations”.

destination as determined by an agreed addressing schema. This implies that in this new model there need not be computing entities inside the network in order to provide services. In this new model, service functionality and user applications alike are located on computing and communication devices outside the network. This demands a packet-based network-access model, where there is no intrinsic concept of a session, but which allows a session to be implemented on top of itself. This can be viewed as user controlled multiplexing.

This schema removes the limitations that restricted the types of services in switched telephony networks. The design criteria mentioned here are among the more important ones that were used for designing the Internet Protocol (IP). The enormous growth of new services that we have seen come about on the Internet since the advent of the Web in the past decade, were enabled by three properties that switched networks lack:

- End-to-end service transparency
- Multiple services per access
- Intelligent and capable end devices

By adopting these properties and this new model, we are now able to fully exploit the computing power and the flexibility of user interfaces in the end-points.

1.1.2 A Voice is heard on the Net

Regarding enabling technologies for service growth, in recent years we have witnessed radical changes in the telecommunications industry. These changes were initiated around 1993 by the growing popularity of protocols such as HTTP, which allowed users to link and share information via distributed computer networks. Initially this was used largely by educational and research organizations. This linked and distributed information store was nicknamed ‘the world-wide web’. The network underlying it was the Internet. The Internet was not recognized to be anything more than a medium for distribution of information or software. Early experiments (1975-1992) with multimedia sessions between end-points in computer networks [26,42,120] were laughed off by the telecommunications industry. But by late 1995, these earlier research results had spawned commercial hardware and software demonstrating the feasibility of making audio calls across the Internet, this commercial hardware and software was viewed much more seriously by the telecommunications industry. The realization that the transmission of real-time voice over packet networks could be more cost-efficient than using fixed amounts of bandwidth in switched networks, and that in addition this allowed enhancing existing services with the more powerful user interface of computers, thus improving the (telephony) services, has precipitated a huge research and development (R&D) effort. Among the outcomes of this R&D effort are protocols for multimedia communication and the establishment of a new ‘Voice over IP’ industry. Voice over IP (or VoIP) commonly refers to the protocols that are used to transport voice (and/or video) and the associated necessary signaling by means of the Internet Protocol (IP). Initially this work was based on H.323, an ITU-T standard for real-time voice and video communication over packet networks [178]. During the past two years, the IETF's Session Initiation Protocol (SIP) for establishing multimedia sessions [59] has become enormously popular due to its open communication model and much lower complexity (as compared to H.323). However, SIP can be said to be “older” than H.323, as SIP is based on research results, preceding H.323, dating from the first two IETF “audiocast” experiments in which live audio and video were multicasted from the IETF meeting site to destinations around the world, which subsequently grew into the MBONE [25].

During this time, we have seen a radical increase of bandwidth, not only in the backbone of the Internet to terabits-per-second, but even access networks [36], where users now are currently moving from 56 Kbps to up to 400 Mbps [182], and bandwidth is expected to continue to increase at the same rate in the years to come. This has clearly removed any doubts regarding whether multimedia communication is possible on the Internet over fixed networks. These trends are so obvious that the
media industry is developing applications that are far more demanding in bandwidth than anything ever imagined for personal communication services in switched telecom networks. Industry is rapidly adapting services and content to the format and possibilities that the Internet offers.

1.1.3 Convergence is fueled by VoIP

The scope of telephony is thus being extended to the Internet and other networks that run the Internet Protocol (i.e., IP-networks), and we are currently seeing a proliferation of development and business efforts related to this convergence. Systems vendors (e.g., Ericsson, Cisco, and others) from the telecom and datacom industry are offering communication solutions based on IP-networks that integrate VoIP with traditional services in the telephony networks. A strong trend is to move all existing functionality to IP-networks. Network operators and ISPs (e.g., Telia, Telenor, and others) are currently conducting extensive field trials in the market place. However, with the availability of VoIP-technologies on the Internet they are not only in the position to build applications that do everything that was already possible in switched telephony networks, but also combine this with what is possible on the Internet, thus giving these applications expanded functionality. Unfortunately, to a large extent the industry is simply occupied with looking into ways of using this technology to rationalize their infrastructure, by transferring familiar services along with their existing business models to these new networks, and not yet exploiting the new services as they are not well aligned with the existing business models.

1.2 Wireless Industry at the Crossroads

1.2.1 3G and Beyond

The model for personal communication has been switched mobile telephony using fixed amounts of bandwidth reserved for voice communication between two parties. Its protagonist GSM has been very successful at implementing this model and achieving a world-wide penetration of 250 million users before the year 2000. The enormous popularity of the Internet has caused us to look for access to Internet through any network and the GSM industry responded with the Wireless Application Protocol, which placed itself between the user and the Internet. Unfortunately this left the user at the mercy of the operator, who controls the WAP-gateway and thus determined what services were available to the user. To make things worse, users are connected over circuit switched wireless access, incurring excessive costs for very limited service. These limitations have been removed in NTT DoCoMo’s iMode2 where users are connected to the Internet by means of packet-switched network access, and where third-party content is not published in the operator’s gateway, but can be sent directly to the mobile as compressed HTML (cHTML). A general-purpose packet radio service (GPRS) is now becoming available in GSM, and the bandwidth will increase in successive generations towards that of Universal Mobile Telephony System (UMTS). Simultaneously, Wireless LAN (WLAN) technologies are available at increasing speeds (e.g., IEEE 802.11b at 11 Mpbs), which can deliver several magnitudes greater bandwidth but with a more limited range. Consequently, we can build mixed systems, where wide-area wireless access (GPRS/EDGE/UMTS) is interspersed with WLAN access (so-called hot spots). However, in light of the fact that we can deliver both asynchronous and isochronous multimedia with end-to-end IP connectivity over all these wireless links, we need to reconsider what services we really will be delivering and also reconsider the basic design of the architecture.

1.2.2 Introduction to “πG”

This dissertation uses the term “πG”, for an economy of words, to refer to a heterogeneous wireless infrastructure of 2.5G/3G interspersed with WLAN nodes, and other methods for wireless access that

---

2. The removal of a basic limitation does not refer to the business model, which is limited to one operator, but how the service delivery leverages the technical components cHTML and wireless packet access via PDC.
may become available (instead of the common use of “4G” to refer to this infrastructure). The architecture of this heterogeneous wireless infrastructure goes beyond the architecture of third
granulation wireless networks (3G), but does not constitute a generation in the common sense, hence \( \pi \) since \( \pi \) is greater than 3 (but not an integer). This choice circumvents collisions with alternative interpretations of “4G” (e.g., comprising novel very high data-rate radio interfaces). Hereafter, when “\( \pi G \)” is mentioned in the dissertation, it simply used, for an economy of words, to refer to the heterogeneous wireless infrastructure of 2.5G/3G interspersed with WLAN nodes that is currently being built, along with other methods for wireless access that may become available. However, this non-claim regarding a universal definition of 4G, does not forfeit other claims made in this dissertation about what I have chosen to denote as \( \pi G \), in that \( \pi G \) offsets new ways of service delivery to users, which will simply bypass existing service delivery frameworks for the delivery of new services, as the properties of these existing frameworks have limited service growth for many years (e.g., IN, TINA, etc. — see further Section 3.1).

Two novel aspects characterize the resulting heterogeneous wireless infrastructure of 2.5G/3G interspersed with WLAN nodes. First, this conglomeration of heterogeneous networks provides end-to-end IP connectivity over wireless links. In addition, aside from mobility support for devices (e.g., Mobile-IP) and support for direct anonymous public access, it is a “stupid network” scenario, where the network only provides packet transport, and therefore it is an “operator-less” network with respect to services. The network scenario for \( \pi G \) networks as outlined in the introduction appears to be straightforward, but we need empirical results in order to know how easy and cost-effective it really is to provide wireless access to the Internet to end-users. In addition, this network has to provide access to existing services of which telephony (voice) is the most important one. We are also in the position to provide new mobile multimedia applications that take advantage of the fact that end-points have computing capabilities, and through sensory capabilities can have knowledge about the context of users. Furthermore, the “operator-less” service scenario also implies that the mobile users and devices that participate in communication over \( \pi G \) must become smarter, i.e., they must be able to respond to a wide range of events:

1. Other users, mobile devices, and communication resources may become “visible” in an ad hoc fashion, either by proximity or actively communicating.
2. Entities (users, mobile artifacts, and virtual objects) may exchange events that range from simple invitations to join a session all the way to manipulations of shared virtual objects.
3. The communication conditions will vary between and even within access networks. This is especially the case where wireless communication is concerned. Applications must be able to act reasonably given knowledge of the situation.

An application architecture for such adaptive, mobile personal communication has been described in [84] featuring mobile agents, VoIP/SIP-enabling multimedia applications, using end-to-end IP communication between users over wireless links.

The questions are thus how to provide easy public access to these services to end-users, understand what functionality is needed, and to be able to demonstrate the feasibility & cost-effectiveness of the architecture, and its potential regarding applications. My hypothesis is that such a heterogeneous wireless infrastructure beyond the third-generation (\( \pi G \)) is both practical and feasible, and that it can be enabled to bring such adaptive mobile multimedia applications to end-users. These were our purposes for building an experimental \( \pi G \) wireless test bed infrastructure, and verifying our application architecture by prototyping. Our results are described below. Later in this chapter we shall discuss the requirements and network aspects for use of mobile service knowledge in eXtensible Service Protocol (XSP)-enabled communication.

1.2.3 Convergence?

Therefore, what we need are alternative service architectures that better utilize the properties of the Internet Protocol (IP), which allows any host to provide a service to any other host. Thus, services can
and should be moved out to end-devices. In addition, unlike switched telephony network services, which are call-centric — i.e., session invocation (granting a resource request corresponding to an “off-hook” event) and session termination (relinquishing a resource corresponding to an “on-hook” event) — we need to generalize this model and allow any event to be input to our model of communication. The Session Initiation Protocol (SIP) for establishing multimedia sessions [59] allows us to move the point of integration for multimedia service integration out to the end-points. With the emergence of multi-mode terminals we can now choose between different bandwidth wireless access networks ranging from moderate bit-rate GPRS to high bit-rate IEEE 802.11b wireless LAN (WLAN) extensions to Gigabit Ethernet networks [83].

Consequently the wireless industry, which is standardizing and designing solutions for the delivery of multimedia services over wireless access to Internet in 3GPP, is at the crossroads of either reinventing extensions to telephony or (realizing that it already has telephony services) supporting wireless access to both voice and other multimedia services thus extending the Internet, but without requiring up front compatibility with all the supplementary telephony services of switched networks. As was pointed out in section 1.2.1, other wireless technologies are available and therefore services, if denied appropriate support in 3G will utilize other access networks to serve end-users.

Thus, the term ‘convergence’ inappropriately suggests an end-state of the Internet having merged with telephony networks, be it via direct access to Internet through the use of modems at either end of a switched speech channel between the end-point and an access server that is co-located with a switch or mediated access via another protocol and a gateway, cf. WAP 1.0 [212]. In both models, the principle method of access is bundled with the service that the network provides via which access to the Internet is delivered. In reality, we are seeing IP networks being used as a cost-effective technology replacement through which pre-packaged telephony services are tunneled (e.g., GSM on the Net\(^3\)) to end-devices and to end-users who are content with a familiar telephony service. However, at the same time new services (as exemplified in section 1.3.2) are delivered to users over these same networks along with traditional Internet access. In this case, the principle method of access to the network is no longer bundled with any particular service. Hence, if any ‘convergence’ is taking place, it is telephony services migrating to the Internet. Consequently, it is not a matter of if this migration to Internet is going to happen, or when, but how fast the migration that is already in progress is going to take place.

1.2.4 Towards a (Multimedia-Enabled) Wireless Internet

A more productive way of looking at things is to recognize that the Internet Protocol (IP) removes a number of limitations that previously restricted the type of applications that we were able to create in telephony networks. IP allows applications to benefit from the fact that end-user devices are now able to multiplex multiple services over a single link. By virtue of the fact that IP enables multiplexing of services, we are now able to build new interactive applications, running in the end-stations, which can combine both voice and data simultaneously. Furthermore, IP hides the underlying network from the applications, which enables us to locate services anywhere that there is IP-connectivity and this allows us to radically simplify the intermediate network(s). Even more importantly, its allows for the rapid deployment, at low cost, of new services that can be used instantly by those who are connected to the IP network, thereby creating a whole new service industry.

Therefore, we should instead focus on the development of packet based access points and mobile devices that are able to communicate via IP over a packet radio access network by simply running IP over the radio link protocol. The Internet Protocol (IP) aided by significant processing power in the end stations permits multiple applications and services to access the transport medium simultaneously, resulting in an architecture where hosts have roles equal to servers. Services may be

3. GSM-on-the-Net (GSM@Net) is a discontinued Ericsson solution for business communications, offering IP-based voice (fixed and mobile), data, and multimedia services over corporate intranets.
triggered by any event, not just a (telephone) call [75].

Furthermore, the cost to transmit digital information end-to-end is dropping dramatically, due to a
tremendous increase in the available bandwidth, not only in backbone networks, but also in access
networks for both fixed and wireless access. The price/performance of end-user electronics is
dropping while there has been a tremendous increase in computational power. Thus, we are in the
position to exploit personal communication and mobility in order to create new applications and
services that go far beyond what telephony systems have been concerned with and were able to
accomplish.

1.3 Motivations: Towards Ubiquitous Computing & Communication

On the motivations for this research and its central hypothesis.

1.3.1 Advances in Mobile Computing

As we have seen, until now the industry has been very much occupied with extending and
leveraging familiar services, while improving the cost-effectiveness and/or the user interface to these
services. On the other hand, by relaxing or removing the limitations that were built in because of the
switched telephony networks, we are in the position to combine packet based networks with
technology advances in other areas thus enabling new services. First of all, the available computing
power in consumer electronics device has gone up dramatically, while the production cost has been
steadily dropping. Power management has improved to the extent that pocket organizers are now
equipped with color-displays, infrared communication, WLAN connectivity, and a 200 MHz
StrongARM processor, and soon we will see devices being shipped with GPS receivers in order to
determine the position of the device. Research laboratories have shown communication devices
equipped with other sensors, such as humidity, acceleration, and temperature, all miniaturized to the
extent that we speak of wearable computing [106]. Secondly, short-range radio technology not only
enables eliminating wires between devices, but via ad hoc communication also allows us to extend
our range of communication to other devices. The advances in short-range radio communication, ad
hoc communication protocols, the dramatic increase in computing power in devices, and on-going
miniaturization, combined with ever increasing available bandwidth in both the backbone and access
network, challenge our current conception of what constitutes personal communication.

In particular, advances in mobile computing research have shown that personal communication can
greatly benefit from what has become known as context- and location-aware computing. Such
applications take into account aspects such as:
  • who is communicating, with whom or what,
  • the context which that person or object is in, and
  • the location or position of the person or object (since it could be an intelligent mobile
    object).

These applications utilize devices such as active badges, sensors (direction, speed, temperature,
position - both relative and absolute), and more. This awareness has been shown to be highly relevant
to achieving much more meaningful communication.

Examples of research regarding context-aware devices are Active Badges developed at Olivetti
Research Laboratory (ORL) [152] and the SmartBadge [106] developed at HP Laboratories, the later
includes also numerous sensors and substantial computing capability.

1.3.2 Scenarios, New Models Of Communication

Our current conception of personal communication is firmly rooted in a century of telephony - with
a small extension via a host of supplementary services. Unfortunately, little can be done to further
improve their usability or extend them, as these services (not surprisingly) are entirely call-centric and
the devices have very limited interfaces. On the other hand, the future world of personal communication can, and will be, much more sophisticated. Mobile devices equipped with sensors (e.g., infrared, location, temperature, acceleration, etc.) are able to determine the user’s context or the communication conditions in which they are used and thus adapt their mode of operation accordingly (e.g., reduce/increase their power, delay their transmission, etc.). Sensors can also be used to allow us to recognize electronic devices or physical objects and interact with a virtual representation of this object augmented by a certain amount of intelligent behavior. We are beginning to build systems in which sensory devices and wireless communication enable mixed-reality experiences (blending virtual and physical objects). This in turn enables us to operate in the feedback loop of virtual or physical computational objects and vice-versa, thus implementing what is being referred to as humanistic intelligence. But this is not science fiction: already today these technologies are being successfully applied by industry to create a spectrum of immersive multiplayer games (e.g., Quake), where groups of users interact with multimedia in virtual spaces. Consequently, beyond simple leveraging of the cost-effectiveness of IP-technology with VoIP, by reinventing telephony on the Internet, we can envisage new modes of communication that allow people to interact wirelessly with worlds consisting of virtual or real objects by bodily movements, where voice (and other multimedia communication) does not require any conscience action from the user in comparison to real life.

What we are attempting to achieve is enabling users as participants in a communication space. If users are allowed access to communication services where no distinction is made between voice and data, we enable services and communication artifacts such that a user becomes a participant in a communication space. Thus, by adding VoIP to a context-aware communication infrastructure, we enable the creation of adaptive applications where voice is no longer an add-on externally provided through some other media. This approach to ubiquitous computing [154] is fundamentally different from systems that mediate between a person and media (e.g., active walls at MIT Medialab [156]) or that either guide media to a person (e.g., Active Badge enabled Telephony System [136]) or vice versa (e.g., Cyberguide [105], Stick E-Notes [19], and the GUIDE system [34]).

The immersive participation that has been enabled by my approach may be experienced as a communication space of ubiquitous computing. It can now also be complemented with the creation of mobile artifacts that bring embodied virtuality, where real-time media is an integral aspect. For instance, we may be given a networked multi-sensory and multimedia Furbee™ (toy), that upon relaying our identity to its home location starts an intelligent (audio) conversation with us.

On the other hand, we do not have to go to these levels of user interaction and vigilance to discover real needs and uses for moving intelligence from the network out to the mobile devices. Joe Mitola’s doctoral dissertation on Cognitive Radio demonstrated how mobile devices can learn to optimize their behavior relative to their communication context and to how they are being used [111]. In light of the advances in using end-to-end mobile multimedia over (wireless) IP links, this approach may be extended to apply to the application level, so that applications can learn how to adapt their behavior in the end-devices relative to what they discover about communication conditions, available network resources, and user behavior—specifically:

1. by responding to our physical context, we can augment physical reality and create mixed- or augmented reality services [12].
2. by responding to communication conditions, we can adapt our communication to behave optimally, hence when we experience scarce or greatly varying bandwidth, we still are able to experience a meaningful service [76].
3. by responding to the communication context of the device and/or user, we can envisage the instantiation of spontaneous and ad hoc communication involving various resources based on the properties of the parties involved in the communication.

The above scenarios show the potential of what can be enabled in terms of applications, in light of the removal of some important limitations concerning personal mobile communication that were inherent to switched telephony networks. In summary, the hypothesis of this dissertation is that the
1.3.3 Specification and Negotiation of Service Behavior

As we have noted earlier, mobile devices are becoming extremely computationally capable, they are able to not only generate and process the multimedia, but are also able to generate different visualizations of the multimedia (per user rendering), gather information about the user’s context (e.g., using position, direction, acceleration, or other sensory information from either the user or the environment) or communication conditions, and make decisions to adapt its own behavior. Therefore, user carried mobile devices, resources, and (potentially intelligent) virtual objects should be able to connect to each other and auto-configure their communication. Thus, we need mechanisms and an open service architecture that enable these entities to accomplish this without having to rely on unique services that are present in a single type of access network, nor should we have to rely on a single operator or even the presence of a network to setup these service. Thus, we allow users to use ad hoc applications. Ultimately, we wish to create a plug & play Internet, allowing users to connect to virtual spaces that can be correlated to physical spaces and be able to jointly observe and manipulate objects, while conversing with others in real time. For example: user A joins a mobile group and visits some physical locations, where he or she leaves location-dependent voice annotations with notification triggers for others in the group to discover. Subsequently, User B passes the same location and finds the note & plays it.

Once the mobile computing infrastructure is in place, sensors, mobile devices, resources, and users must be able to negotiate their communication. JINI [195] and UPnP [209] are two available technologies that enable parties to register and use services in mobile networks, and thus to some degree support the dynamics of ad hoc communication that we wish. However, it can be argued that they neither scale well (for wide spread use in public networks) nor do they address negotiation of services beyond simply pattern matching service profiles. Consequently, we must look for a means, which enables sensors, devices, resources, and users to engage in communication with a level of understanding of the task in question, in order to be able to negotiate their communication.

From the above scenario we may infer the need for generic protocols for creating relationships between devices and users, a generic data model for describing the properties and service behavior of devices and users, a data store combined with inferencing mechanisms, the data model of which matches the generic data model for describing properties and service behavior of devices and users. Making these profiles accessible on the Internet, and allowing every profile to refer to and relate to other profiles determines their semantics. This is the approach taken in this research and presented in this dissertation. Specifically, any person, artifact, or place may be represented by a software object, which allows it to be inspected, to demonstrate independent behavior, and act based on the occurrence of external events, and thus by virtue of these properties it has an agent. These agents can be located in the things they represent or can be in entirely different places requiring that other parties understand the (correlation) naming and localization schemas and protocols.

1.3.4 The Hypothesis and Approach

In our model for personal communication, the services of the past will migrate and become an integral part of a communication space on the Internet. We may then cease to perceive them as services. This communication space will interact with the physical world by means of context & location-aware technology. In fact the physical world can be represented and visualized, and this allows us to interact both with each other and the things that we know, within this communication space. At the same time, we can interact with entities in this communication space that have no existence in the physical world.

Users can exchange knowledge pertaining to communication services or resources, such that other users are then able to use them. The dynamic behavior that the personal Active Context Memory notion of Smart Spaces [43] can be extended to the scale of mobile networks and integrate its existing services.
brings about constitutes the communication application. Furthermore, it can be enhanced with learning capabilities that improve the communication over time or via interactions with others. Services may now be created by “ad hoc” aggregation of resources that are dependent on the communication context. The proposed personal Active Context Memory is able to instantiate such services depending on the communication context. The eXtensible Service Protocol (XSP - see Section 7.1) allows mobile users to ad hoc join in (or create) an infrastructure and automatically negotiate services with minimal user intervention and a minimum of shared service knowledge. No central server or control points are needed, as we use event routing to disseminate Mobile Service Knowledge. Consequently this architecture scales well and enables us to adapt and enhance our communication in new ways as outlined in the introduction.

The advantages of our approach are:

• Creators (vendors, operators, or third parties) of services are able to design and ship service components without the necessity of standardizing all service concepts as XSP allows descriptive meta-data carried by the object to resolve queries from another object during service negotiations.

• Service creators may ship service components and use the personal communication context to emit data. After collection, this data allows data mining based on user behavior and can be used to predict the success of new applications and service components.

• We are no longer obligated to pre-design application behavior. Rather such behavior is emergent in our model, in that we can design the personal communication context memory to learn from communication situations and thus exhibit rapid adaptive behavior (both to the user and communications).

• Applications or aggregate services may be instantiated in an ad hoc fashion; they may involve various devices from the current communication context. This is particularly valuable now that very short-range radio networks are becoming available (e.g., Bluetooth [164]). The result is that an ad hoc composition of various devices can be used in a single application.

• The architecture permits off-loading of computation to the end-points by servers in the network provided that the necessary re-synchronization mechanisms are in place. This also allows for off-line behavior and roaming in mobile networks, which are not connected to the Internet.

This architecture was created by introducing a personal Active Context Memory, into a context-aware mobile communication infrastructure with wireless access, as referred to above. Using this, we are able to create open-ended adaptive applications for personal communication. Secondly, the XSP constitutes an open vehicle for negotiating communication between parties, and enables mobile networks to evolve into large-scale open systems of cooperating devices and resources. Third, the combination of the Active Context Memory and XSP makes it possible to build these rich applications at low cost and to rapidly introduce them (as compared with traditional telephony services). Finally, there is no implicit need for centralized network management of these applications.

1.3.5 Revisiting the Role of VoIP

In the new service architecture we need to incorporate multimedia communication, hence take advantage of IP-multiplexing for multiple services and events. IP also enables applications to be agnostic of the underlying access mechanism while delivering a specific service. We argue that we can only successfully cost-effectively deliver the proposed new personal communication services [69,70,71,72,73] if multimedia services are delivered (multiplexed) using a single link (via IP). Since without this multiplexing we would need to have an additional infrastructure for the communication of events for service negotiation, etc. - this would both increase the infrastructure and the cost & power consumption of mobile devices. Thus, VoIP is a key element of this new service model.
1.4 Specific Problem Statement

Given the new possibilities that have been presented in this chapter, we are thus confronted with questions regarding:

- How should we realize and manage applications and services for personal communication? (see Sections 4.6 through 4.10, Section 8.2 and 8.3, and exemplified in Chapter 9)
- What range of (important) applications and services are enabled by the approach? (see Section 4.3 and 4.5, Chapter 9)

In turn, this gives rise to a series of related questions:

- What requirements do these new applications and services impose on the network — especially when wireless access occurs? (see Section 2.3, Chapter 8, and Sections 9.4, 9.7, and 9.8)
- What are the implications on the design of clients and wireless terminals? (see Section 2.4, Subsection 8.3.3, and Section 9.7 through Section 9.9)
- What are the new models that are brought about for provisioning these new applications and services? Or to put it in another way - who will be delivering what to whom? (see Section 2.5, Section 2.6, and Section 8.2)

Regarding the above-mentioned approach we also need to answer the following, in order to assure the feasibility of the scenario (see sections, 8.3) and further demonstrated in Section 9.7 and chapters 8 and 9:

**Scalability**

Does it scale? How do we get it to scale for large numbers of users? (see Section 4.5, Subsection 8.3.15)

**Management**

What are the minimal properties or requirements for a management-less solution? (see Subsection 8.3.21)

**Step-wise Introduction**

Can it be introduced step-by-step without extra demands on the network? see (Subsection 8.3.16)

**Network Economics**

What are the network economics? (see Subsection 8.3.17 and Subsection 9.6.4)

In subsequent chapters I will give answers to the above questions. In light of this new model of communication, the design goals for the communication network and its components are to support negotiation of these new services between end-user and service provider, without requiring the assistance of the network operator or network access provider. Even more importantly, we should allow any user to become a service provider whenever applicable (e.g., live multicast from an event). The network must of course deliver the content as necessary to meet the requirements. This overall design goal can be divided into the following parts:

**Deregulated Access**

End-users are allowed to connect to the Internet via any network. Trust relations between an end-user and any visited network can be negotiated via a trusted third party, thus enabling unbundling of the local loop in combination with anonymous Internet access [66] — see Section 8.2 and Section 9.5.

**Device Mobility**

End-users are allowed to roam within a network or between networks, implying that any on-going sessions must be retained. This requires the automatic assignment of a network address to the mobile device and redirection of the on-going communication to the new address, either by making this redirection transparent as a network service (e.g., Mobile-IP), or combinations of a transparent network service and mobility awareness on the application level (e.g. SIP and Mobile-IP) [153] — Section 4.9.
Service Mobility
A service that is available to the end-user in one network should also exist for the user in another network. Not only should this service remain reachable in terms of addressing, but it should also exist for the user in the new network and operate in the new context (e.g., a print service should discover available printers in the new environment) — Section 4.9.

Personal Mobility
End-users are allowed to move to other networks and remain reachable using the same address, not only for voice communication (e.g., personal numbers in fixed or mobile networks), but also for any other application. End-users can switch between different end-devices, but should be reachable by the same address — Section 4.9.

Adaptive Communication
The end-user’s mode of communication adapts itself based on events (e.g., not just calls, but also based on sensor input, such as spatial data, bandwidth, content availability) to user context or communication conditions. This implies that users can benefit from storing and reusing service data in order to learn how to respond optimally to communication events (see Subsection 4.4.7 and Subsection 4.4.8).

Ad hoc Application Negotiation
The negotiation of communication between users, resources, and virtual objects should require only minimal shared service knowledge, implying the necessity for a general-purpose service negotiation protocol (see Chapter 7).

1.5 Overview
This research began with an investigation of the feasibility of delivering mobile multimedia over wireless links with end-to-end IP connectivity. Analysis of performance and effectiveness in service delivery were identified as important properties along with identifying the key global and local properties required of future networks - the ability to adapt to the changing context of the user and communication is a driving factor for the success of new applications. Not only is this property true of the link layer, but even more importantly, recognizing that adaptation is fundamental, allows us to build new adaptive services.

In order to facilitate this adaptation to context, the point of service integration is moved out of the network and into the end-devices. A novel approach, called a ‘Mobile Interactive Space’, was presented that provides for interaction between representations of people, devices, and resources, connected by ubiquitous communication, thus constituting a reference model for the service architecture. The model is called Adaptive Personal Mobile Communication. In addition, an analysis of the implications of this approach for new business models was presented.

Following an analysis of different service architectures for delivering these services and the necessary properties of an open model for describing, managing, using, and exchanging service components, an open service architecture for multi-service, multi-event communication for the integrated control of applications and service components (including mobility and VoIP) in end-devices, using (mobile) agent technology, was defined and explored.

The research thus created new network and system level models for building applications in which the new requirements and design-rules can be mapped. This resulted in the proposal of new components and mechanisms: an eXtensible Service Protocol (XSP) and an XML-based specification language (Mobile Service Knowledge: MSK), for describing, managing, using, and exchanging service components in the ‘Mobile Interactive Space’. Furthermore, I propose the use of a personal Active Context Memory attached to the global identity that sets trigger conditions for communication
such that it allows users to interact with services in a meaningful way or when a new resource is encountered it can be interrogated for its capabilities and then used, through negotiations using the eXtensible Service Protocol (XSP).

Through the creation of applications with these properties along with rapid prototyping, in local and metropolitan area wireless test beds in order to assess scalability, deployment, management & network economics of these solutions, and through analysis - the contributions of this new approach have been characterized. These applications and analysis will be described in subsequent chapters.

1.6 Contributions and References

This section states the main contributions of my dissertation, referring to the criteria that were put forward in conjunction with the problem statement, and with pointers to where these contributions are shown and described in this dissertation. It is important to note that the contributions are the results from a research project and that more aspects need to be addressed before such a system might be rolled out for mass use, that have to do with the design of target systems (e.g., GPRS, 3G, etc.), rather than the principles which this dissertation has produced.

<table>
<thead>
<tr>
<th>Contribution Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Necessary Properties of New Communication and Related Work</td>
<td>Chapters 1, 2, 3</td>
</tr>
<tr>
<td>Feasibility Investigation</td>
<td>Chapter 9</td>
</tr>
<tr>
<td>Analysis of Business Model Implications</td>
<td>Chapter 2</td>
</tr>
<tr>
<td>Open Service Architecture Framework</td>
<td>Chapter 4</td>
</tr>
</tbody>
</table>

Necessary Properties of New Communication and Related Work

In view of the new possibilities that VoIP-technologies have created [Chapter 1, Chapter 2] examined the limitations and fundamental constraints that restrict services in existing telecom services architectures [Chapter 3]. An overview is presented concerning related work regarding technologies, components, and systems and what has been accomplished in different attempts to remove these limitations and enable new communication.

Feasibility Investigation

Chapter 9 shows the feasibility of multi-service personal communication that is multiplexed via end-to-end IP connectivity over (wireless) communication links. addresses the software architecture (including requirements) of the mobile device. Chapter 9 and [72,74,85] examine the minimum requirements for some key components of the new service architecture [Chapter 4], concerning the performance and effectiveness of multimedia delivery, and minimum requirements for VoIP (in particular Wireless) communication. This analysis demonstrates that the adaptive capabilities of the agents can be used to further relax the requirements.

Analysis of Business Model Implications

Chapter 2 examines the impact on existing business models as well as enabling new models. This approach allows services to move out of the network. Given that present access networks are becoming more diverse and deregulated this enables users to move between networks (perhaps anonymously) while preserving their communication and services

Open Service Architecture Framework

Chapter 4 presents a new model for an any-event multi-service communication model in the Internet (Mobile Interactive Space - by allowing us to encapsulate a communication identity with service knowledge for all communication entities in agents) as well as an open service architecture introducing new system and network levels (in particular, services are separated from the signaling protocols, that make the services available) which is able to fully exploit the potential of an any-event, multi-service communication model for creating adaptive mobile applications. the new service architecture, further detailing the new system and network levels, this is followed by the discussion of the design and interaction of protocols.
Chapter 5, Chapter 6, and Chapter 7 present in turn key components of the new service architecture, further detailing the new system and network levels, the design, and interaction of protocols.

**Adaptive Service Behavior Components and Mechanisms**

Given the service architecture, Chapter 5 examines the properties of Mobile Service Knowledge as a vehicle for the intelligent and adaptive behavior, the internal dynamic properties and mechanisms, and interaction with the external XSP level.

Chapter 6 discusses the requirements and design of an Active Context Memory, the purpose of which is to reason about Mobile Service Behavior in order for the agents to facilitate intelligent responses to communication events.

Chapter 7 describes the design of an eXtensible Service Protocol which enables peer-to-peer ad hoc communication sessions and application negotiation between entities, enabling them to negotiate and extend services and invoke communication potentially aggregating multiple devices in this communication.

**Network Requirements and Properties**

Chapter 8 outlines the network requirements and properties for managing mobile service knowledge and making it available to users (provisioning by mobile operators). Its also addresses the scalability, network economics, and step-wise introduction, along with the management of the approach, meeting the requirements of heterogeneous access, and deregulated service provisioning. Furthermore, an analysis of the network properties is presented of the Open Service Architecture demonstrating that it is scalable, “management-less”, and allows for step-wise introduction.

**Range of Enabled (New) Application and Services**

In Section 4.3, a vision of a new communications paradigm for these new services and applications is presented that are enabled by the new service architecture, VoIP, and wireless networks. Chapter 9 characterizes these applications are in further detail and using examples of enabled communication by discusses prototypes that implement different scenarios and presents a wireless test bed, as well as the components of which these prototypes were built. These results prove both the feasibility and practically of the theoretical results of previous chapters.

In summary, these results demonstrate that ubiquitous computing technologies applied to creating smart-spaces can be scaled to the size of mobile networks while providing user-centric computing & communication involving multi-service communication that is able to respond to intelligently to any event negotiating & extending its service repertoire without requiring predefined common service concepts that must be stored in the networks.

The presented open service architecture is thus ideally suited for deploying services in a deregulated fashion (i.e., there is no requirement for the access and service operator to be one and the same) via heterogeneous (wireless) access networks as part of the Internet.

Furthermore, the service architecture enables the services to be moved out to the mobile devices, while simultaneously introducing mechanisms that enable application providers and network operators to query the stores of mobile service knowledge for empirical data of how the service are used (in order to learn about their relative success, or even simulate service usage to predict their performance or success).

This provides an important answer for what applications 2.5G and successor networks can support and characterizes the protocols, service architecture, and properties of mobile service knowledge needed to enable such applications.
1.7 Summary of my published papers and other documents

This dissertation is based on the following published papers and one work in progress [86]. For each of them I describe my contribution(s).


Note: My contribution to this paper was a model of a service architecture for Internet applications, and identification of its necessary properties: multiple services (both voice and data) over a single link, a communication identity for each of the communication entities, and the separation of service behavior from the underlying communication. I also showed in what way network architectures for cellular networks, along with their business models have to be rethought, by arguing for the viability of new wireless multimedia communicators replacing the mobile phone.


Note: This publication describes a communication space with public and personal devices and identities that are able to communicate through this services architecture. I specifically contributed a “teleliving concept” (stemming from the fact that we can be represented by a communication identity on Internet). Secondly, a method for how a personal communication identity can be attached to services for personal communication or vice versa, as well as a model for service behavior.


Note: In this poster I described how virtual spaces of ubiquitous computing can be enhanced by VoIP to offer multimedia communication spaces, based on the service architecture, and what new services are enabled by this approach. This was exemplified by a Web Contact Center. Furthermore, I showed the implications of the service architecture upon how services and network access will be brought to end-users, as well as the implications for current business models.


Note: In this paper I contributed the requirements, the verification, and proof of the viability of the proposed wireless multimedia communicators [69], replacing mobile phones. In light of Christian Olrog’s complementary results of how to create a direct link for IP over GSM, I further detailed the impact for network architectures and access technologies, and in particular third generation wireless networks.


Note: My contribution to this paper was the concept and model of a communication identity augmented with intelligent communication behavior and showed how it could be integrated in the service architecture. In addition, I presented preliminary requirements along with a model for knowledge representation of service behavior. Furthermore, I showed how this method enables new uses of services for instantiation, data mining and learning.

*Note:* In this paper I further detailed the requirements, the verification, and proof of the viability of the proposed wireless multimedia communicators [69] and showed how these come into play when third generation wireless networks are rethought and built as IP-networks with routers and radio access points. This paper expands on the results of [72].


*Note:* In the licentiate thesis I presented the open service architecture, which is further detailed and extended in this dissertation, enabling services to be moved via the Internet and out of the network to the mobile devices, in particular facilitating the integration of VoIP and data-services in multi-service communication, for intelligent responses to *any* communication event. In the thesis I investigated the feasibility of this architecture, examined what new classes of applications were enabled by this approach, and presented examples of the latter.


*Note:* In this paper, my contribution was to show how smart delivery of multimedia content involving agents running in the mobile, the base station, and the content provider allows us to dynamically adapt the application and network behavior to each other in order to meet the criteria for specific applications. This allows us to avoid interference between these delivery mechanisms, while maximizing the range of services and the number of users.


*Note:* In this paper, I presented a method using XML-based protocols and agents operating on a distributed tuple-space in order to enable common awareness of these events in applications for mobile users, without the necessity of these events being defined or announced in advance. The paper also further characterized classes of personalizable applications in interactive mobile spaces, which are enabled by the approach.


*Note:* In this paper I characterized the properties of heterogeneous wireless packet access networks labeled as fourth-generation wireless networks (referred to as ΠG in this dissertation) and show the feasibility, ease, and cost-effectiveness with which these networks can be built for provisioning new applications in Mobile Interactive Spaces in our urban wireless test bed, describing the application architecture for Mobile Interactive Spaces together with how the eXtensible Service Protocol allows users, mobile artifacts, and (potentially intelligent) virtual objects to engage in spontaneous communication.

Note: Given my co-author’s work in security on Local Area Networks, I showed how Fabio’s results could be incorporated in the open service network architecture for Mobile Interactive Spaces.


Note: In this paper, I presented in detail a novel eXtensible Service Protocol (XSP), which is part of the open service network architecture, enabling us to use and deal with spontaneous application level connectivity. After examining its properties, I showed how XSP enables ad hoc mobile applications, as well as enabling adapting applications to the conditions and context of the communication.


Note: In this paper, my contribution was showing how adding service knowledge meta-data to the multimedia content (creating so-called Smart Media) interpreted by agents located at specific points (routers, access points, mobile devices) enables us to further free resources for the delivery of streaming media to mobile users. I also examined the minimal requirements for delivering these services.


Note: In this paper I described the application architecture for adaptive mobile communication for Wireless Internet featuring an eXtensible Service Protocol that has the necessary properties for scaling a plug & play smart space to the size of large mobile networks. I further showed the feasibility of the architecture and cost-effectiveness by describing the implementation in a wireless testbed of a prototype adaptive mobile-aware media-player.


Note: In this paper I listed various limitations in 2.5G & 3G cellular networks regarding services, and contrasted this to a characterization of heterogeneous wireless packet access networks labeled as fourth-generation wireless networks (referred to as 4G in this dissertation). I showed the feasibility and cost-effectiveness of building such networks and provisioning of adaptive mobile multimedia applications by extending the emerging broadband infrastructure via wireless LANs. Furthermore, I presented a service architecture to negotiate adaptive mobile multimedia communication, with minimal shared service knowledge, which enables applications to adapt to and make optimal use of the heterogeneous mobile infrastructure. I also contributed the results from building a mobile-aware media-player, which was extended to take into account the user’s context.


Note: In this journal publication, I characterized the properties of service architectures in relation to the steps taken in successive generations of wireless communication networks for personal communication and examined the limitations imposed by them on services in wireless
Thus presenting a novel service architecture for open communication in wireless Internet, describing its necessary properties and evaluating its merits. Finally, I described my experiences building application prototypes utilizing augmented reality in our service architecture, within an urban wireless testbed.


Note: This journal publication further details and consolidates the results from papers [72,74,84]


Note: In this paper I show how XSP removes the limitations of UPnP and enables negotiation of services between end-points for multi-device application scenarios, without requiring advance knowledge of these hosts. XSP leverages results of the Semantic Web effort [9] for specifications of semantic information governing the relations between mobile devices, users, and virtual objects, and integrates it with the service architecture framework for adaptive personal mobile communication that it is part of.
This chapter presents an overview of existing and new methods for gaining fixed or wireless access to Internet, followed by a discussion about the issues and requirements with respect to wireless QoS for delivering multimedia to mobile users, concluded by examining the improved capabilities of mobile devices for achieving new modes of operation employing multimedia, and utilizing wireless access. A final section investigates the impact on models for access and service provisioning, and examines to what extent this impact already has occurred.

2.1 Fixed Access to Internet

2.1.1 Dialup Access

Switched telephony access, be it copper lines (PSTN, ISDN) or cellular (GSM), allows us to do signaling for setting up communication channels with fixed bandwidth, which is advantageous for an operator who can charge per time unit for the access to and usage of these channels. However, packet-oriented transport is advantageous for network operators to use in the backbone of the network for optimal utilization of the available bandwidth, as packets are only sent when there is a need to transmit new content, rather than occupying fixed bandwidth, irrespective of need. Unfortunately, a customer does not benefit directly from this technology replacement in the backbone. In order to gain access to (multimedia) services, the user has to resort to dial-up connections.

An early solution, for combining telephony services with dialup Internet access, shown in Figure 2-1, was created by my research group at Ellemtel and later turned into a product (PhoneDoubler™). A Voice Gateway (VGW) is co-located with the Access Router and Modem

![Figure 2-1. PhoneDoubler™ Solution](image)

1. PhoneDoubler™ is a trademark of Telefonaktiebolaget L M Ericsson. Patent Pending.
Pool (AR). The Voice Gateway also carries out the signaling needed for the Voice over IP (VoIP) via terminals located at A and B and translates the signaling need for B’s side (IP) and A’s circuit-switched telephony. Party B has both telephony and Internet access, but via the gateway so does A. While there is no upper bound to the available bandwidth, nor lower bound as to what is actually being used in B’s case, A is limited to 64 kbps. As for the quality of service, no degradation of speech quality or perceivable latency (less than 200 msec.) is measured given a proper dimensioning of the IP network [74]. PhoneDoubler demonstrates the imbalance between potential versus delivered bandwidth and services, as there is ample room for transmitting other data traffic in addition to good-quality voice over the communication link to B, while the underlying connection oriented access only transports voice. PhoneDoubler also shows the advantages of multiplexing Voice over IP & other data traffic over the single dialup connection to B in terms of enabling new services. In order to avoid circuit switching in the access network we must look at other alternatives for delivering greater bandwidth at moderate costs to households.

2.1.2 xDSL

Recent research has lead to a number of products classified as Digital Subscriber Line technologies (xDSL), which can support bandwidths up to 55 Mbps downstream and 2.3 Mbps upstream. This technology was used in field trials [199], between the access-point and nearest exchange, where Internet traffic is separated out. The success of this technology depends to a large extent on the price/performance relation in terms of delivered bandwidth, as well as cost of operation for the ISP. Its primary benefit is that it enables the reuse of the copper “last mile”.

2.1.3 ATM

All field trials that assume user services to be based on ATM-access have either failed or repositioned ATM as a networking technology [199]. Mapping QoS parameters from RSVP onto ATM (where QoS control was the single reason to try to use ATM) is very complicated and has not lead to any successful practical results. ATM, if it has a place, will be in the backbone as a transport mechanism.

2.1.4 Dynamic synchronous Transfer Mode (DTM)

DTM has been labeled as a competitor to ATM. Field trials building high capacity IP-based networks based on DTM have been successful. However, it remains to be seen if DTM is a viable access technology for IP in the face of cheap, near ubiquitous 10-baseT, 100Tx, and Gigabit Ethernet. As the capacity of single DTM channels is 512 Kbps and IP concepts map much more easily on DTM [62] than on to ATM we may expect that DTM is well suited for use as a backbone for delivering real-time traffic with certain QoS-requirements.

2.1.5 Packet-oriented (Ethernet/IP)

A competing access technology is to provide Ethernet access directly to homes. Field trials by Telia and Ericsson in Stockholm have shown that 100 Mbps Ethernet Internet Access can be offered using Gigabit Ethernet backbones at low cost [177].

2.1.6 Cable Networks

Deployment of connectivity over cable networks, including wireless cable, Local Multipoint Distribution Service (LMDS) or Multichannel Multipoint Distribution Service (MMDS), is a viable option. Telia currently offers wireless broadband access service with bandwidths up to 25 Mbps in

---

2. E.g., The Broadband Trial Integration (BTI): one of the ACTS 3rd Call projects. The objective of the BTI project was to develop and demonstrate a concept for improving the quality of Internet services through integration of Internet protocols and ATM.
Stockholm, Malmö, and Gothenburg.

2.1.7. Electrical Power Distribution Networks

The power industry is enhancing their infrastructure to deliver information and voice services to households. We may expect the bit-rate of traffic for consumers on these networks to increase dramatically in the near future [44].

2.2 Wireless Access to Internet

While IP has been widely accepted for transporting multimedia over fixed access networks, there has been quite some disagreement about what it takes to provide interactive multimedia applications over wireless access networks. This disagreement is largely due to the packet loss over wireless links and the presumed scarcity of bandwidth.

In the local area, Wireless LAN access has been available for some time and end-user devices and applications are available in the market that address the needs of office-environments (for example the Symbol Technology “phone” which uses a wireless LAN infrastructure [196]). With the growing popularity of wireless LANs (WLANs) that currently offer speeds up to 11 Mbps, and which are expected to increase to beyond 50 Mbps in a year’s time, even more bandwidth will be readily available, thus further reducing any remaining limitations on the type of multimedia communication that we may experience using wireless communications.

In addition, there is presently a major investment in third-generation wireless networks for wide-area access; the telecom industry realized that packet-radio access to Internet via wireless networks (beyond the second generation, - e.g., GSM) can accommodate not only services for sharing of information and media, but can also support real-time communication between users. The basic premise is that here too, lots of bandwidth is needed to provide multimedia services. However, experiments with multimedia applications based on VoIP built on top of mobile computing devices with Internet access over both local area and wide-area cellular networks, have resulted in strong indications that this premise is plainly wrong. I will describe these experiments and measurements below.

2.2.1 Satellites

Satellites are frequently used to deliver high bit-rate connectivity. Tele2 (a Swedish operator) launched a hybrid Internet-access service using MTG ViaSat in spring 1998 targeting small businesses, where customers could download with rates ranging between 200 and 300 kbps, using a dial-up connection for the uplink. Today Tele2 uses the Sirius II satellite with bit rates between 250 and 800 kbps. The user is merely required to add a satellite-card (to their PC) that connects to their 60 cm satellite dish. Low Earth Orbit (LEO) satellite networks, such as Teledesic are planned, that when fully deployed promise to deliver 2 Mbps uplink and 64 Mbps downlink per user with global coverage. Globalstar and Iridium, are geared to telephony services and low data-rate data services and cannot provide high data rates. In May 2000, EarthLink announced that it launched satellite Internet access across the continental U.S [183]. EarthLink promises download speeds of up to 400 Kbps and uploads at speeds up to 128 Kbps. [114] shows that VoIP over satellite is not only desirable, but also feasible. In addition, investigations regarding a Gigabit Wireless Access Network [140], show that communication via air-borne equipment utilizing very high data-rates, is only in its infancy.

2.2.2 Local Area Wireless Communication

Wireless LAN (IEEE 802.11) supports communication between mobile hosts without the assistance of a base station. Home RF [171], and the OpenAir initiative [184] provide further alternatives. IEEE 802.15 Wireless Personal Area Networks [174] (Bluetooth) is currently working to produce a standard for co-existence of local wireless communication technologies, in particular Wireless LAN (IEEE 802.11).
2.2.3 Short-range Wireless Access (Bluetooth)

Another available but yet unproven technology is Bluetooth [164]. It offers a short-range radio link, which supports interaction with up to 8 devices simultaneously and a total address space of 64 devices (where one is the broadcast address) and a maximum bit-rate of 64 Kbps for voice and higher data-rates for data. Even with its limited capabilities Bluetooth in combination with (for instance) JINI [195] or UPnP [209] is a powerful enabler for pervasive computing applications.

2.2.4 Cellular networks

GSM offers a data channel with data rates not exceeding 9600 bps. This rate is further reduced by protocols that are not aware of the characteristics of cellular medium access and control protocols. For instance, TCP (which is in turn is used by HTTP) will insist on re-sending complete IP-packets if a frame on the radio link level gets lost. Alternative solutions have been proposed, such as the Wireless Application Protocol (WAP) [212]. However, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video. A General Purpose Packet-radio Service (GPRS) has been developed that is able to offer IP-connectivity. Unfortunately, this protocol is not suited for isochronous real-time content, such as voice or video.
all available bandwidth, design of the user interface enables middleware to mediate demands for bandwidth and apply application based traffic shaping. In most other cases (except those involving video and images) there is enough bandwidth so that bandwidth by itself is no longer an issue.

2.3.2 Resource Reservation

Voice has different characteristics in comparison to that of bursty data traffic. Thus resource reservation is often necessary in cases where the available bandwidth is limited. However, as Section 2.3.1 showed bandwidth is not a problem for voice traffic, therefore we should not introduce and complicate our solutions by requiring resource reservation schemes (such as RSVP).

2.3.3 Latency

Latency in fixed networks is a matter of dimensioning the network and not a real issue. Network latency in local area networks is of the same order of magnitude, i.e., tens of milliseconds or less, while wide area networks typically incur latencies in the order of 50-100 msec (or even up to a few hundred msec depending on how the network is dimensioned). However, latency is a problem where Internet access is provided through (digital) modems over switched cellular network access such as in the (worst) case of GSM-data [72,74,85]. The only logical solution is to remove the circuit switching and transport IP directly over the Radio Link Protocol (RLP) in GSM or Radio Link Control & Medium Access Control (RLC/MAC) in GPRS [169]. The remaining latency is due to the interleaving, this enables a trade-off between packet-loss (i.e. to increase robustness) causing retransmissions and latency. As modern speech codecs are able to deal with heavy packet loss (up to 40%) without significantly lower perceived quality, this interleaving can be eliminated and hence latency need not be an issue.

2.3.4 Robustness

Latency is also important for interactive network games, such as Quake3 Arena and Unreal Tournament, but in this case delay is not correlated to congestion problems as the end-points send at low data rates (in the range of 13 kbps) for status updates of local objects. Instead latency is mainly caused by the (radio) network interface and to a lesser extent by network delay (see Section 2.3.3). On the other hand, it is important that the packets arrive, otherwise synchronization problems will occur. For interactive network games arrival may not be critical as continuous updates of the objects’ (absolute) locations are sent. This is normally handled by TCP or by UDP along with an application-specific handshake protocol. However, the link level may also provide this service. The interleaving that GPRS does increases the probability that a packet is not lost, but on the other hand the user pays a penalty in increased latency, even if the packet is not lost. The GPRS Application Alliance has tested the currently most popular network game, Unreal Tournament with a GPRS simulator with good results [188]. Using only one timeslot, the latency (about 300 - 800 msec) at 26 kbps incurred by GPRS interleaving does somewhat adversely affect players’ performance and appreciation of this application, but not in a critical way. Note that this latency is higher than the round trip network delay, hence retransmission would provide a lower average delay.

2.3.5 Encoding, Decoding, and Transcoding

Encoding, decoding, and transcoding all incur a delay. As end-points are able to negotiate their communication (i.e. chose matching encoding and decoding), this removes any need for intelligence and transcoding in the network. Speech coders typically incur less latency than the network - latency incurred by many speech coders is much less than 40 msec. Unfortunately, latency is cumulative.

The coding/encoding algorithms (CODECs) also dictate the upper bound of the perceived QoS of multimedia streams (such as speech, video, or audio). The lower bound is dictated by the percentage of link packet-loss that the CODEC is able to tolerate before its performance suffers severely. Modern
Adaptive Personal Mobile Communication

codecs (e.g., Voxware RT-24) can tolerate up to 30-40% packet loss, with no additional latency.

2.3.6 Header Compression

IP without header compression introduces a non-negligible overhead. Particularly in the case of low bandwidth wireless access where bandwidth should not be wasted this is especially harmful. Ericsson and Nokia have proposed and successfully demonstrated a robust algorithm for removing headers from IP packets carrying voice ROHC [17,37] that has demonstrated considerable better performance than CRTP [27]. Thus IP headers do not take up significant bandwidth.

2.3.7 Summary

In summary, achieving sufficient QoS for voice and game updates is not a fundamental problem, but rather a design issue that can be dealt with in each case by properly dimensioning the hardware, software, network access, and transport protocols. Similarly in order to support images and video we can achieve the necessary QoS by proper dimensioning.

2.4 (Mobile) Devices

This section examines enhanced capabilities in end-devices and increased capabilities for utilizing wireless access to model new modes of operation for end-users.

2.4.1 Enhanced Capabilities

Users of telephony services are experiencing the effects of a strong trend towards incorporating more complex and powerful functionality into mobile phones. Also mobile computing devices, such as laptops and PDAs, are evolving rapidly in various ways. The production costs of electronics are decreasing rapidly to the point where electronic communicators are becoming consumer products. The computing power is increasing rapidly (e.g. StrongArm 200 MHz processors are common). We are seeing new choices of operating systems (Epoc, PalmOS, Linux, Windows-CE, etc.) some with smaller footprints than others but that are able to run powerful applications. Furthermore, minimizing power consumption and doing careful power management, which combined with improved battery technology, ensures that mobile applications are available to users over reasonable periods of time. In some cases, radios have been integrated into various devices, such as the Nokia 9110.

2.4.2 Utilizing Wireless Access

Even more importantly, there are now devices available that use a single wireless link for simultaneous voice and data communication, such as Symbol Technology's Netvision® Data Phone. Due to the availability of short-range radio, in the future, we may even expect to wear these computing & communication devices (attached to our clothes or as “jewelry”) and use them in concert via a body-area network. Due to the increased processing power we are able to do encoding and decoding of speech at the end-station, thus enabling applications that incorporate multimedia capabilities for interactive communication, such as videoconferencing [75]. The available processing power can also be used to adapt the mode of communication (e.g., when this end-station is enabled to detect who is using it, the speech coders can for instance use speaker-dependent knowledge to optimize voice quality) or respond to communication events. Consequently, only the instantaneous available bandwidth, processing power, and battery life limit us in shaping our communication.

2.4.3 Mobile Users

Most mobile computing equipment (such as notebook computers) now have multimedia capabilities, due to increased processing power, the inclusion of soundcards, cameras, and USB-interfaces to other devices. Wide-area and local-area wireless access to the Internet has sufficient bandwidth to accommodate simultaneous access to multimedia and web-based applications,
as shown in my papers [72,74,76]. Voice can be integrated, transparently, in the user interface, without
the necessity for the network to be aware of the content of the traffic. This wirelessly extends the
mobile computing device to the ensemble of devices within range of the short-range radio link.
Furthermore, mobile users may have additional use for devices that connect the mobile computing
equipment that they might have access to via short-range radio-links (e.g. Bluetooth [164]).

2.5 Changes in the Communication Industry

This section examines the consequences of multimedia service delivery, brought about by ample
(wireless) bandwidth and low cost of infrastructure (with consumer electronics prices for end-user
equipment), and its effect on roles in the communications industry, resulting in a ‘wild-west scenario’.
It goes on to show that services cannot be packaged with the network access. Hence third parties can
and will deliver the new services. Thus we are heading for a complete deregulation of the
communications industry, where the old services (voice) will come practically for free when bundled
with the new services.

2.5.1 Introduction

Currently, the penetration of wireless networks and Internet access in the Nordic countries exceeds
50% of the population (75% for wireless); similar trends are perceived in other regions of the world.
The telecom market is highly deregulated with a number of operators able to offer services. When
Internet access is offered, mechanisms are available (e.g., DHCP, protocol filtering, etc.) to regulate
the traffic, but there are no straightforward mechanisms for regulating the services that are delivered
on top of IP. Any attempt to implement such regulation defeats its own purpose by putting up
roadblocks for service growth. Therefore we may expect that traffic regulation will be exercised in
order to exercise a sufficient level of QoS and security, but restricted so as not to defeat the purpose
of the providing wide-spread (and wireless) Internet access, which is to create a market-place for
services that are able to reach customers in whatever access network (broadband Internet, GPRS,
WLAN, UMTS) they may happen to visit. The Internet42 field trials [177] by Telia and Ericsson have
shown that large amounts of bandwidth can be offered at a fraction of today’s cost, and have enabled
another, new company (Bredbandsbolaget) to exploit this possibility [172]. Access to the Internet can
be offered in a completely deregulated fashion. Private organizations such as housing co-operations
may acquire connectivity to backbones, and distribute this connectivity further at no or low cost.

With multimedia communication possible with only modest requirements in terms of bandwidth,
there is no reason for large mobile telephony operators to package network access and services.
Mobile phones are becoming a commodity; while consumer electronics (such as personal data
assistants) are also being equipped for wireless access. In this chapter, we are therefore looking at a
future wide-open deregulated market where users will be flooded with consumer electronics
connected wirelessly to a deregulated flat-rate Internet that is used for global communication. When
this happens, the large mobile telephony operators will have to adopt a datacom model of doing
business or find themselves overtaken by others who understand the new way to do business.

The alternative is in the short term to lobby for regulations, which will slow down the introduction
of these new services by alternative operators.

4. iamasia DataByte, 26th April, 2001: 40% of Hong Kong home Internet users now connect via broadband.
5. According to Telia (December 1999), all of Sweden’s municipalities and 3000 main communities are connected in a fiber-optic
network, within reach of 91% of the population and 95% of all businesses. According to their annual report for 2000, Telia
alone had orders for broadband Internet access units to 822 000 households - out of a population of 8 862 311 (source:
Statistiska CentralByrån (SCB) November 1, 1999).
6. During 2000, the growth rate for residential high-speed Internet access in the US exceeded 230 percent, and is predicted to
7. Press release 3/8/01 A Brand New World, regarding a module combining GSM and WLAN for a Pocket PC.
2.5.2 Deployment of Services

New demands for services favor short development cycles and a very short time to market (TTM). Hence, the requirements on the new infrastructure are as follows:

- Services should be deployed as is, without the necessity to test them for interaction due to using common network resources.
- Thus the services need to be designed to be easy to field.

2.5.3 Telecom Model

Telecom services are notoriously hard to introduce due to:

- All services having to be integrated with intricate and complicated charging support systems that will host multiple operators.
- Service interaction testing is very costly as all services use common resources in switches and are very much network-aware.

This is entirely due to the fact the telecom networks offer only switched access and the basic premise is that whoever puts up a network can only get paid for making connections via this switched access.

2.5.4 Datacom Model

Internet access with ample bandwidth can be delivered to end-users. Services and applications can be designed to be largely unaware of the network and be written once, but accessible everywhere in the network. Almost anyone can put up a service and sell it to whoever has the requisite connectivity and desires the service. Network operators or Internet Service Providers (ISPs) can only charge for the bandwidth used or based on value added due to functionality that speeds up response time or enhances the service in some (but still largely transparent) way.

2.5.5 Cost-effectiveness

In summary, the cost of delivering the service is decreasing both in terms of the total cost of the infrastructure that is needed to access the service and the cost of creating and deploying the service. Due to the high volume of datacom equipment sales the price/performance is an order of magnitude lower for a given bandwidth than the telecom equivalent.

2.5.6 Consumer Model of Computing Capable End-User Devices

End-users are now in the possession of commodity priced multimedia capable computing devices. This has created a demand for new services that both telecom types of networks and their successor broadband testbeds have failed to deliver.

In the remainder of this chapter, I describe the potential of services using IP-access, and that IP-access is fast becoming a commodity and widely available to households and professional users. Individuals have become used to dedicating a large portion of their household budget to computing (one or more computers) and communication (already over 57% of the Swedish households are connected to the Internet, and there is over 80% penetration of mobile telephony and Internet access. The telecom

---

8. According to the IDC worldwide shipments of PDA units was close to 9 million. NPD INTELECT reported the PDA market to have grown 161 percent during the year 2000 (i.e., to roughly 15 million units).
9. A comparative estimate of the number of cellular phone users in the US in the year 2000 is 112 million according to Merryl Lynch (source: eMarketer 2001), which through technology substitution and in increasing number will become part of a mobile Internet.
10. In May 2001, 57.6 per cent of Swedish households were connected to the Internet, according to a press release “UK continues to lead Europe and the US for visitors to e-commerce websites”, NetValue S.A., http://www.netvalue.com, June 2001
industry has responded by introducing WAP-telephones, whereas the datacom industry has brought us PDAs with wireless access, both of which fall within the budget of an ordinary household in the industrialized countries. Market trials by NTT-DoCoMo in Japan with a similar idea, mobile phones enabled with an “i-mode” (Internet Access), to access third-party services, revealed a huge consumer interest for accessing digital services over this wireless Internet\textsuperscript{13}.

Integrating the multimedia capabilities of these devices along the lines of this dissertation is the next logical step as shown in Symbol Technology’s NetVision\textsuperscript{®} Data Phone \cite{196}. This has far-reaching implications for the business models of the actors in the communications industry. We will therefore see that actors with an existing business (mobile phone operators) will look for solutions that buy them time. Newcomers (cable TV, power utilities, transport companies, etc.) will not hesitate to bypass the earlier telecom solutions and make money by selling access to their customers to those who are looking for new opportunities (third-party services, amazon.com, etc.). Thus, what we are looking at is a consumer-oriented communications market.

2.6 Business Model Implications

2.6.1 Packaging the service with the network

We have seen the reasons why end-user services cannot be packaged with network access, as this would require a service architecture that imitates the bad aspects of the telecom model. It would also require us to put up large servers with equally complicated charging support systems, thus hampering service creation. Telecom operators who have gone into the ISP business with such ideas have failed to make money. As a result, and with the conclusions from the previous chapter as background, we expect that the telecom model of deploying services will go away. That process has already started \cite{127} and will accelerate.

2.6.2 3d Party, Consumer-model

Banks \cite{191}, insurance companies \cite{207}, power industry \cite{210} and transportation companies are all embracing this consumer focused model of offering end-user services. Those who have an infrastructure (e.g., power industries, transportation companies, etc.) see an excellent opportunity to enhance this infrastructure with IP-connectivity at very low cost. Being able to deliver new services to their existing customer base \cite{44} by leveraging their existing investment in infrastructure, they are able to expand their businesses into new markets. Those companies who have an existing business are able to gain leverage from this new infrastructure by the fact that these well-known services can be made available to any customer and at any location with Internet connectivity. Any effort to adapt their existing services to the new medium are more than paid for by the cost reduction brought about by having to support fewer customers via physical office locations. These are important drivers, and we expect that any service that lends itself to be adapted to the Web, will be on the Internet. VoIP is an important enabler, as most non-trivial services require human interaction (at least occasionally). These companies will not rest until nearly everyone is connected so they can be assured that their potential customer base is maximized.

Another emerging consumer model is well illustrated by the rise of vehicle telematics (e.g., OnStar\textsuperscript{14}) where communication adds new value to existing consumer products (in this case, cars).

\begin{itemize}
\item[12.] The Strategis Group predicts mobile data subscribers in the US to grow from 5 million in 2000 to 172 million in 2007. In addition, mobile data will be instantly available to GSM subscribers (450 million worldwide in the year 2000 - source: GSMworld).
\item[13.] According to Newsbytes, 01 May 2001, Japan's mobile phone penetration approached 64 million units of which 20 million signed up for iMode in the first two years.
\item[14.] http://www.onstar.com/visitors/html/ao_easy_operation.htm
\end{itemize}
2.6.3 Infrastructure Development Phases

These end-user applications will initially not be focused or even include isochronous real-time services such as voice or video, because the majority of consumers have dial-up Internet access with varying performance in terms of latency and bandwidth. Consumer pressure hand in hand with third party companies that demand an improved infrastructure for delivery of their services will lead to substantial improvements in the Internet infrastructure. In 2005 we may expect a greater than 50% market penetration, in the industrialized world, of on-line households with at least 1 megabit/s bandwidth for their Internet access.\(^\text{15}\)

An exception to this is satellite based downlink based services (such as XM Satellite Radio, Direct TV, Direct PC, etc.), which are focused on high-speed media delivery, primarily non-interactive audio and video. These operators are primarily providing popular multimedia content.

In light of this, there is a certain amount of idiocy involved in current attempts by the telecom industry to re-invent itself. For instance, CCS7 was ported to IP in order to be able to use IN-nodes on the Internet to set up VoIP-calls and thereby exporting its already failed model of service deployment. The telecom industry is trying to maintain its position by migrating its traditional functionality, but I believe is just loosing time. The solution is to adopt a new model.

2.6.4 Cellular Pitfalls

While the battle is nearly lost for fixed networks, the telecom industry would appear to be in a strong position for wireless (cellular) networks. However, as has been shown in the previous chapter, there are no technical reasons for the telecom industry to continue to maintain its model of deploying services. The position that vendors and operators of cellular networks have (e.g. GSM) is entirely based on economics and market dominance. In order to maintain this position, new solutions have to be adopted, such as a packet radio service (GPRS) and more bandwidth (EDGE, W-CDMA).

Unfortunately, the traditional telecom industry advocates a model of service delivery and deployment that already has failed for fixed access, because that is the only model they know. The telecom industry feels secure because they assume they are the only ones able to understand and able to deploy wireless wide-area access. Fortunately (for the consumer) this assumption is wrong because wireless Internet access can be easily offered anywhere customers will regularly be, without the necessity of cellular networks. As was described in the earlier chapters, it is simple to do so, by for instance, extending the broadband Internet access of communities and housing cooperatives with wireless LAN access points.

2.6.5 Wireless Wild-West

In the near future, we will therefore, exactly as is the case for fixed access, see a deregulated communications infrastructure for wireless access, offering low-cost, flat-rate per-maximum-bandwidth Internet access that allows us to communicate using multimedia content. When we add to this commodity priced, multimedia- and computing capable end-user equipment and short range radio to extend the scope of user devices wirelessly to resources nearby, we are facing a whole new model of doing business:

- **Deregulation**
  
  Network Operators will simply provide Internet Access and charge based on bandwidth, plus service enhancements like caching. Users are able to shop around (perhaps even in real-time [18]) and negotiate access as they wish.

- **Consumer Market**

---

15. In their 2001-06-01 press release, Powernet announced that 450,000 households in Sweden will be provided with public broadband wireless and indoors fixed Internet access via their cable network that has coverage throughout Sweden (http://www.powernet.se/main.asp?id=nyheter&nyhet=38#nyhet38).
Businesses will sell access directly to end-users applications to enable them to use connected things - e.g. toys, communicators, and home appliances — these are only some examples of connected equipment which can be taught to act in concert [56] for professional use, entertainment, etc. See further Section 3.9.3. This field is wide-open and may be labeled “wireless wild west” [170] to illustrate the vast expanses of communication and computing applications that it opens up.

2.7 Chapter summary and conclusions

This chapter has presented an overview of existing and new methods for gaining fixed or wireless access to Internet, followed by a discussion about the issues and requirements with respect to wireless QoS for delivering multimedia to mobile users, concluded by examining the improved capabilities of mobile devices for achieving new modes of operation employing multimedia, and utilizing wireless access. A final section investigated the impact on models for access and service provisioning, and to what extent this already had an impact.

Venture capitalists, operators, providers, and system vendors earlier assumed that WAP 1.0 would provide a sufficient basis for this “wireless wild west” of services (and business opportunities), and that investing in a 3G infrastructure alone would provide a sufficient migration path for further expansion. These assumptions were proven wrong, but more importantly our analysis shows that we must rethink the architecture and components for delivery of these services, in order for these services to proliferate, and also what will be their impact on existing business models.

New Application Providers (e.g., power industries, banks, transportation companies, AOL.com, Amazon.com, etc.) are very much interested in leveraging the services that they already might have by gaining access to a large new potential customer base via the new infrastructure. In addition, they will be very interested in also expanding their access to a potential customer base by fielding new services. Furthermore, equipment providers (e.g. copying machines, household appliances, etc.) or even the toy industry, will find itself to be a beneficiary by producing networked products that have content or associated services on-line. In this case, interfacing services that were uniquely tailored for access to a specific type of communication (e.g. telephony) are increasingly less relevant. For instance, banks were able to move their services to the Internet (web) with very little effort. Thus, these new application providers will be very keen to maximize their interaction with customers and explore all the possibilities that the new service architecture gives them.

This change in business models calls for a new service architecture along with application support that makes intelligent use of the infrastructure and meets the requirements that new dynamic multimedia services, delivered by third parties, will put on it. Chapter 4 presents this architecture and examines its design, after examining existing solutions and related results in Chapter 3.
3 RELATED WORK

The changes mentioned in the previous chapter have profound consequences with respect to how we should build our networks (e.g., GSM, IN, CTI, etc.), in particular coming generations of mobile networks (3G and 4G) - and the choice of solutions for service architectures. Multimedia communication over IP has been addressed in new work in ITU-T, ETSI, and the IETF. We examine how far these protocols and architectures go in solving the problems posed in chapter 1. This chapter concludes by discussing technologies for open & ad hoc discovery and use of communication services.

3.1 Limitations in Existing Telecom Service Architectures

This following sections presents some key existing telecom service architectures and analyzes the reasons why today’s personal communication as provided by these existing telecom service architectures is rigid, unintelligent, and prevents the growth of new services as envisioned in Chapter 1.

3.2 Computer Telephony Integration

A goal of Computer Telephony Integration (CTI) is to enhance telephony services by exploiting the fact that a computer is co-located with a telephony device. The computer contains the application’s graphical user interface and the functionality to control the services that are located in a central switch. The service model is a step forward - from Advanced Intelligent Networks (AIN) and GSM - in that it allows an application that is located near the end-user, specifically located in the computing device, to react to events in coordination with a central service node. However, this delegation of control is severely restricted by the central service node. The user must have an account there with well-defined (and restricted) properties that enable the service operator to do intricate charging. The application cannot easily be extended to integrate new events and additional sources that would enable an adaptive user interface. This is simply because the application relies on a service that comes with the circuit switched network access, and it is implemented in the central service node. We cannot rapidly or easily extend the functionality of the application when we introduce new resources into the user’s environment, since this would require changes to the central service node. The application cannot even adapt to these new resources unless we design such adaptation into the central service node. In this respect, CTI is no better than IN or GSM as a service platform.
Critics might say that end-points could contain the services, but then our earlier comments about IP over ISDN would apply here as well, i.e., the separation of voice and data channels in the end-points does not scale well. A recent development is to incorporate VoIP in the clients and thereby eliminating the restriction of connection oriented network access. However, we might still be restricted in our services by an entity (the H.323 Gatekeeper) in the network whose role it is to control sessions as explained later in section 3.8.1.

In the office environment, Computer Telephony Integration (CTI [5]) solutions use computers to provide easier-to-use user interfaces for such tasks as user status, typing messages, placing, receiving, and forwarding calls. Screen Phones (for households) attempt to improve the user interface and provide easier access to supplementary services. For mobile telephony, so-called smart phones have been introduced that combine office-automation software with CTI-functions in different form factors. Yet another approach is to utilize data communication with dedicated servers or gateways, by means of the Wireless Application Protocol (WAP [212]) in order to create CTI-services in so-called ‘SmartPhones’. These WAP-servers or gateways may in turn use information from the Internet in these services.

In CTI, the service logic is moved out to a PBX and hence is closer to the user. Communication, as packaged by the network access, is focused around sessions. Hence, there is an intrinsic limit to the kind of services that a user is able to experience within such a system. This limitation stems from the fact that the user interface has to be aware of the concept of a call. No matter how we design our user interface, it will require the user’s attention for setting up sessions and this dictates the way user interfaces may be designed. What is even worse, is that the logic of the service cannot successfully be located anywhere other than bundled with the network access, unless we introduce severe limitations in services, along with limits on personal and device mobility. Unfortunately, even customer control and CTI-arrangements do not fundamentally improve the situation as they merely move the location of the bottleneck to the PBX. These limitations explain the lack of success for ISDN and provide a reason for why its successor B-ISDN never took off.

3.3 ATM, B-ISDN

In this section I discuss briefly ATM and (B-)ISDN as successors to analog switched telephony that share the property of being connection-oriented while attempting to incorporate some properties of packet-network access. A short analysis is given in each case of why the connection orientation property seriously limits the services that we can create using these network access methods.

While ATM uses cells as its transport mechanism, the actual protocols used in ATM to setup sessions between end-points are connection-oriented, as the goal was to guarantee a certain quality of service (QoS) as opposed to simply a best effort service. This service architecture copies the limitations of ISDN into a technology designed to provide broadband sessions. As a result, ATM will not come into play in the new services at all. At most it will be used as a bearer of IP, for which other better price/performance (service and switching) technologies are available.

(B-)ISDN suffers from the same limitations as IN and Cellular Telephony, therefore the comments regarding the service architectures in these networks also apply to ISDN. ISDN uses B-channels to transmit voice or data and a D-channel to send signaling packets. This D-channel can be used to tunnel IP-packets enabling end-points to be on-line and set up voice and data channels. However, this separation of voice and data channels imposes another serious restriction on the type of services that we can construct. Either the client or the server needs to integrate events in the voice channel and data channel. This does not scale well, for instance when multiple parties want to join sessions that they discover. Fortunately, this model can be changed using IP-multiplexing of the voice and data services as was described in Section 1.1.2.

ATM, being the foundation of B-ISDN, is a connection-oriented mode of communication, although it sends and routes cells to transmit content. Therefore, it shares exactly the same limitation as other switched telephony access technologies (e.g. PSTN, ISDN, GSM), in that the user applications have
to deal with and are limited by the concept of calls (i.e., connections), which are bound to this network access model. These basic services are integrated in and not be separated from the network access model.

3.4 Tunneling Telecom Services over IP- Networks

Re-inventions of known solutions, such as Call-Centers and Private Branch Exchanges (PBX), based on Local Area Networks (LANs) running IP and VoIP-technology to carry voice, have appeared in the market. There are two prominent examples of wireless access to these solutions. GSM-on-the-Net™ (see also Section 1.2.3), terminates VoIP in the base station and end-users are able to use their existing GSM mobile phones. Unfortunately, this precludes any possible new services, since the user simply sees a GSM service. Symbol Technology, however, uses a wireless LAN infrastructure to bring IP-based services all the way to the handset (IP end-to-end), including VoIP. If an open service architecture is adopted (which is one of the main points of this dissertation) such an infrastructure could be extended to provide the new services via these (wireless) local area networks.

3.5 Intelligent Networks (IN)

In the past, vendors, service providers, and operators of telephony networks have made enormous efforts to provide advanced services, by developing additional functionality in the network, because it supported their business model very well. The services were put in IN-nodes or added through CTI functionality, co-located with telephony exchanges, within the network that these services were allowed to be aware of. We have seen that in this model the basic services cannot be effectively located outside of the network. Therefore, we should not be surprised by the fact that the growth of services in switched networks in the last decade is close to zero. Field trials with broadband networks, based on ATM technology have failed for the same reason. They merely offered a technology replacement for things we already knew or had (e.g. Video on Demand, etc.). In the case of CTI or IN, it lead to stupid user interfaces that dictated and limited what the user could do.

Services in switched telephone networks are invoked by the events that CCS7\(^1\) generates. The events of session invocation and teardown allow a service node in the Intelligent Network architecture to be invoked and respond to the CCS7 signaling generated by the key presses of the user on their telephone. This service architecture is not extendable and seriously restricts the type of services that we can create. The lack of growth in IN-services in recent years, the high percentage of services that are either not used or rarely used, plus the inability of most users to use them successfully is proof that this is true. This architecture has later been generalized in a TINA-framework [206], for the CORBA-based distributed computing platform [151], which is the basis for its IN-control functionality. Even TINA did not enable significant numbers of new services.

3.6 3G Roadmap to Mobile Internet

Even if progress is made towards mobile multimedia communication on the Internet, we need to understand and analyze exactly what limited communication in these earlier attempts in order to avoid making the same mistake(s) again in a new infrastructure. Unfortunately, the wireless industry is in the process of standardizing and building infrastructure for so-called third-generation wireless networks, but is reinventing a familiar business model on top of these networks. They are thus reinventing the traditional telecom architectures for service provisioning, retaining the network operators as the

---

1. Common Channel Signalling System #7
regulators of services.

This section characterizes the service architectures that are used in 2G, 2.5G, and 3G. In addition it characterizes the properties of $\pi G$ wireless networks (see also Subsection 1.2.2). Figure 3-3 provides an overview.

### 3.6.1 2G (Second Generation Cellular)

In 2G (cellular telephony), mobile devices authenticate themselves and the identity of the user while reporting their location to the Home and Visiting Location Registers (HLR, VLR), which thus act as mobility management nodes. Cellular telephony also relies on CCS7 signaling and suffers in principle from the same inherent limitations as IN. Speech or data sessions are based on circuit switching of radio channels. The Short Message Service (SMS) provides a very limited packet data service. Except for SMS, all services are mutually exclusive, i.e., only one service can be used at a time. Additional client software in the mobile device (e.g., for Personal Information Management) may be used to invoke the services resulting in so-called Smart-Phones.

### 3.6.2 WAP

GSM has been extended with a Wireless Application Protocol [212], which currently uses Short Message Service (SMS) datagrams to transfer information between the client and the WAP gateway. The WAP gateway is usually designed to transform Internet content into what the clients are able to display. No real-time content (such as voice) can be transmitted and events in the voice channel and

---

Figure 3-3. Service Architectures beyond Second Generation Mobile Networks
data channel are not easily integrated. It is obvious that the complication of transformation of content and the limited bandwidth available to WAP in its current architecture (i.e., WAP version 1.0; latest is WAP 2.0) seriously limits the services that can be created. Furthermore, the cost incurred to the user to transfer content via WAP and introduction of alternative solutions such as the coming introduction of IP over GPRS, which circumvents the limitations of WAP, indicate that this service architecture will soon go away.

WAP-clients in the mobile device offer a simple interface to Internet content that can only be accessed through a WAP Gateway, which translates between IP and WAP protocols. An Internet web server can eliminate the need for such transformation by publishing pages with Wireless Markup Language (WML) tags. Through the use of WML Script content, other services can be invoked (e.g., sending short messages, invoking calls). By following a specific URL, the user can download and play a video (or video clips) from a media server (ignoring the fact that it would be less expensive to have an express delivery service personally deliver the DVD to the user!). Using WAP in the mobile terminal causes user services to be strictly dependent on the functionality of the WAP-gateway, and thus dependent on the network operator. In addition, the circuit switched network access disallows asynchronous application events; this greatly limits the type of services that can be offered to users in a meaningful way.

However, from a short term-perspective, the vendors and operators of mobile telephony networks are in a position to delay the replacement of WAP (by for instance XML profiles over GPRS), as they have invested in the development of infrastructure and handsets for WAP-services. Most likely these vendors and operators will seek solutions that will prolong the life span of WAP. Even if WAP has not yet gained a substantial market share by porting WAP to GPRS, they hope, it will become a big market, because GPRS will radically reduce the cost of communication of content for the end-user. On the other hand, users expect to have Internet access and be able to use these same services without the same limitations as WAP. Third-party service providers (e.g., Amazon.com, etc.) will find it easier

![Figure 3-4. Access Networks in 3G and Beyond](image-url)
to set up web-sites that dynamically adapt their web pages to the capabilities of handsets with XML-capable browsers; thus they circumvent the extra effort and cost of buying content hosting on WAP- portals, thereby gaining direct access to their customers via the Internet - such an approach has been successfully applied in Japan (iMode [175]). The proposed WAP version 2.0 standard has exactly this approach and is expected to do better.

As we have seen in Chapter 2, Internet access also enables third-party service providers to integrate voice and video, using VoIP, without requiring intervention by a network operator to do the service integration.

### 3.6.3 2.5G

GPRS and EDGE will remove some of the limitations, concerning the type of services that can be offered to users in a meaningful way in the previous section, by offering an underlying packet data service. Mobile terminals authenticate themselves to the Gateway GPRS Support Node (GGSN) and report the location of the user to the Home Location Register (HLR) through the Supporting GPRS Support Node (SGSN). The mobile device obtains an IP-address from the GGSN. There are different traffic classes, thus allowing for combinations of switched GSM and packet-switched GPRS traffic. The current standard for GPRS data traffic incurs considerable latency due to interleaving data (in order to increase the reliability of data transfer) and to allow for per packet establishment of radio bearers (in order to optimize utilization of radio resources). In addition, the operator is still in the position to encourage, if not require, that the mobile device be configured to use servers in the operator’s service network to setup multimedia sessions.

### 3.6.4 Parlay

The Parlay Architecture [186] is based on CORBA interfaces that enable hosting of applications outside of specific networks while accessing resources in other networks, through gateways that are installed by the network operator, making these applications and services available to the user irrespective of what network the user is located in. The Parlay API specifications are open and technology-independent, so that anyone can develop and offer advanced telecommunication services.

A SIP server can be used to setup multimedia communication between end-points. The service flexibility can be further enhanced by adding a Parlay-API [186] to the SIP server in order to execute servlets via a CORBA interface on a web server. A web browser can be used for customer control of

![Figure 3-5. Parlay Framework (left) and Parlay/SIP Prototype (right)](image-url)
the services - in what can be regarded as a Virtual Home Environment (VHE), with integrated interfaces to a Service Control Point (SCP), in order to be able to control legacy services. Scripted mobile code can be sent to and executed by agents that are co-located with an application client in the mobile device [48]. Moving the execution of code to the mobile device has various advantages, e.g. performance, and allows the device to report local states back to the server.

Clearly we can move services between different networks, but only within Parlay domains, but this process is entirely controlled by the network operators. Figure 3-5, based on [38], shows an example of how a simple service using these interfaces can be built. This example was used by the authors of [38] to prototype wake-up calls and location-dependent information push services in a mobile network.

What is particularly important about this example is that the controlling web interface and the application are only synchronized through network-based servers across a network boundary. Mobile code can be sent to the device to enhance user interaction, but the process must be carried out under the supervision of the application servers and requires synchronization across network boundaries. Furthermore, the Parlay interface must be changed each time to reflect capabilities that are present or introduced in SIP [38]. Parlay has these two limiting properties in common with other network-centric service architectures, such as WAP, VHE (see Section 3.6.6), and TINA-C [206].

3.6.5 3G Phase 1

Mobile terminals authenticate themselves and report the location of the terminal to the HLR through the combined SGSN and GGSN (which also assigns an IP address to the mobile terminal). 3G Phase 1 supports real-time and isochronous multimedia (e.g., voice calls) using end-to-end connectivity over wireless links, which are set up using servers in the operator’s service network, in order to negotiate session parameters regarding quality of service levels (QoS). A Virtual Home Environment (VHE) ensures that user access to services is independent of the location of the terminal, and that the user interface is independent of the terminal, for instance using (as in 2.5G) a web interface (HTTP) and Java for customer control of their VHE.

In summary, the service architecture does not differ in principle from the one in 2.5G, and the service architecture offered by an operator of a GPRS, EDGE, or 3G Phase1 network requires that any communication, beyond simple browsing of web pages be mediated by servers in the operator’s Service Network. Negotiation of services and levels of QoS linked to network specific Authentication, Authorization, and Accounting (AAA) and mobility mechanisms effectively eliminates any possibility of importing or exporting services to/from Internet without prior arrangements. While this service architecture makes perfect sense from an operator’s point of view (and follows an established business model), it prevents or in the best case makes it extremely complicated to support the types of communication that we propose. Service mobility between this and other networks can in principle be solved on a per service basis, with adaptations to deal with the specific requirements for mediating functionality in the Service Network. However, we believe this makes services harder to deploy rather than easier!

3.6.6 Virtual Home Environment (VHE)

The Virtual Home Environment (VHE) is a concept for providing personalized service portability across network boundaries and between terminals. The concept of the VHE is that such users are consistently presented with the same personalized features, User Interface personalization, and services in whatever network and via whatever terminal (limited only by the capabilities of the terminal) and wherever the user may be located. For 3G phase 1, VHE consists of GSM services & roaming principles and Service capabilities - see Figure 3-6.

The service capabilities offered are call control, location & positioning, Public Land Mobile Network (PLMN) information & notifications, and bearer establishment. It is clear that any services
created in this platform are unique to this platform and network (e.g. CAMEL, MexE, or SAT [202]), and cannot be moved outside of a 3G network to the Internet. In addition, this architecture is not open to utilizing resources in the end-devices or in the network that have not been made available in this architecture through a published API.

In the Release 5 specifications concerning the architecture of the IP Multimedia Subsystem [204], SIP is used for setting up services. However, the point of service integration resides in the network thus limiting the services as pointed out earlier in Section 3.1.

3.6.7 3G Phase 2

In 3G Phase 2, Mobile IP is used for handoffs and roaming between 3G and other networks; hence user mobility is no longer controlled by the (3G) GSN nodes. Mobile terminals authenticate themselves to AAA-servers via the integrated SGSN and GGSN node (IGSN), which also acts as a foreign agent for mobile-IP, thus assigning a visiting IP address to the mobile terminal. The IGSN reports the location change to the home agent of the mobile terminal and forwards the AAA information to the HLR for charging purposes. This AAA and mobility scenario enables the mobile to negotiate communication with resources outside of the 3G network(s) without intervention of servers in the operator’s Service Network. Naturally, the operator can offer support for different levels of QoS and even differentiated charges, but the fact remains that the services are negotiated end-to-end, and not simply inside the operator’s Service Network. However, mobile terminals are required to have detailed knowledge of such support services, which may differ between networks and may change over time. Thus, we need a means to describe shared knowledge of these support services and also a means to automatically obtain such knowledge in order for services and mobile devices to migrate between networks. This is one of the design goals of the eXtensible Service Protocol (see Chapter 7).

3.6.8 3G Service Delivery Framework

The Service Delivery Framework in Third Generation Wireless Networks (3G) is currently evolving through the stages shown in Figure 2-4 and detailed in Figure 3-7 based upon [204]. This approach is open in principle to the provisioning of services outside of the home network. However, the provisioning of services is integrated with detailed arrangements for QoS and charging done in these respective networks allocated in the Call Server Control Functions (CSCF) nodes, raising suspicions that these nodes prevent end-nodes to acquire full control of session initiations, i.e., restricting our ability to move the point of service of service integration out of the network into the end-points.
3.6.9 QoS Support for Mobile Users

In response to the need to better support the delivery of multimedia to mobile users, different proxy-based solutions have been proposed either based on user preference driven filtering (e.g., graphics) such as Web-on-Air (an Ericsson product offering), or based on meta-data descriptions allowing the proxy to send only portions of the multimedia that are relevant to the user preferences or correspond to the user input, e.g. Universal Media Access [65], or agent assisted negotiation of QoS for mobile users [32]. The architecture in [32] is very interesting relative to the purposes of this dissertation, however, as it proposes a mobile-aware proxy, the solution is restricted and can only be seen as part of the more general service architecture framework, that is presented in Chapter 4.

3.6.10 Summary 3G Roadmap

Beginning in subsection 2.3.5, different service architectures for 2G (WAP), 2.5G (Parlay), and 3G Phase 1 (VHE) were presented [203], which share a common property of making network based services available to mobile devices within a network. Removing this network boundary, i.e. locating our services on the Internet, also allows us to move these services out to the mobile device, to the resources, or to any (virtual) object on the Internet. Therefore the service architecture - which is presented and discussed below in section 5.6 and later - should not only be open in the sense that the previous ones said they were (i.e., making the interfaces public), but rather anyone or anything can at anytime publish a service for or use a service to or from anybody else [79].

What does seems to pose a serious problem for developing new applications is the call model inherited from fixed telephony into today's cellular networks, which incurs initial delay for setting up sessions and imposes separate voice and data connections — resulting in the same application limitations that have troubled ISDN. Call models for multimedia communications are currently being applied to access in third-generation wireless networks. Just as ATM has proven unnecessary for the fixed network, circuit switched allocation for voice-calls is unnecessary for the wireless access network.

3.7 πG

In this dissertation, πG refers to wireless access to Internet via packet access provided by a heterogeneous network consisting of for instance WLAN (e.g., IEEE 802.11b), GPRS, and W-CDMA and where there is no longer a direct relationship between the method of acquiring network access and the method for getting service (see also [33]). The choice of πG reflects that it concerns an architecture beyond that of third-generation wireless networks (since π is greater than 3) but that it does not involve a new generation of wireless networks in the common sense (since π is not an integer; see also Subsection 1.2.2), thus avoiding collisions with different interpretations that circulate in the telecommunications industry of what constitutes 4G — e.g., radio interfaces for providing very high-speed wireless access beyond IEEE 802.11a or HIPERLAN2.

Since multimedia services can be delivered with end-to-end IP connectivity over wireless links, this
Adaptive Personal Mobile Communication

40 Adaptive Personal Mobile Communication

allows us to extend all existing voice services to these networks. So-called ‘hot spots’ equipped with wireless LAN (WLAN) extensions to the Internet are becoming available, and today provide us with even higher bandwidths (e.g. 11 Mbps in the case of IEEE 802.11b); for example Telia’s HomeRun system [201], corporate WLANs, and “semi-public” WLANs.

This unregulated deployment of wireless infrastructure is particularly important, since broadband Internet access is being provided in a rapidly increasing number of public locations (hot spots) and in urban areas even homes. The provisioning of broadband Internet access is being installed and/or provisioned by power companies, transportation companies, housing co-operatives, joint-ventures of municipalities, etc., all of whom have a radically different business model than traditional telecom vendors and operators of cellular networks. Extending this packet-switched infrastructure with wireless access points, such as IEEE 802.11b Wireless LAN is straightforward. In addition, mobility solutions, such as Mobile-IP, and IPv6 are available to provide the necessary scalability that accommodating millions of users and devices will require.

Furthermore, solutions for direct access to Internet (i.e., not requiring an existing subscription, but rather a direct settlement, e.g. with E-cash) are available [66]. In fact, such operators simply provide IP-access, they do not necessarily even need to do authentication, authorization, and accounting (AAA), since they get paid directly or indirectly.

Consequently, users with mobile devices can, in principle, use any service from any third party, without any intervention by the operator that provides the network access. It should be noted that attempts to limit the customer’s choice by incumbent operators have been found to violate the EU’s competition laws.

Thus, the properties of πG are such that it provides users with (1) multimedia over end-to-end IP (wireless) links with (2) high-bandwidth, between (3) multiple, heterogeneous, access networks, and with (4) direct access to the Internet and thus end-to-end IP-connectivity to (5) third-party mobile multimedia services, without the need for prior subscription for Internet access with these access network operators.

3.7.1 Business Model Independent Service Delivery Framework

Petri Jokela’s paper regarding providing anonymous Internet access [66] combined with results from Phillipe Charas [33] regarding policy enforcement allow us to create a business model independent service delivery framework in which we avoid potential problems due to remaining unnecessary restrictions due to QoS negotiations and charging (Figure 3-8).
3.7.2 Beyond Circuit Switching and Call Models

Faced with the unsuitability of the (above) existing service architectures that are available in switched networks (or its re-invention in packet-networks), our only choice is to remove circuit switching and call-models as a basis for network access to services. This choice must be made to enable a whole new service paradigm that we refer to as a communication space. IP combined with packet-access allows multiple services and transparent connectivity over a network that has no inherent knowledge of the services; this is a prerequisite for making this new paradigm possible. What is needed is a new service architecture that is well adapted to the Internet. This architecture must be able to take advantage of wireless access and a multitude of consumer devices that can be made to act in concert in an ad hoc fashion. An end-user will be empowered by a personal computing and communication environment in this architecture, instead of being enslaved and restricted by service accounts that network operators or service providers have decided for them — see Figure 3-8.

3.8 Internet Multimedia Architectures

3.8.1 ITU-T's H.323

H.323 is an ITU-T standard for real-time voice and video communication over packet networks. Traditionally, H.323 has been used in order to provide multimedia sessions over ISDN networks. During the past two years it has become the de-facto standard on the Internet for VoIP. The following roles and entities are present in H.323:

Gatekeeper A Gatekeeper sets up, monitors, and negotiates QoS parameters for multimedia sessions between end-points.

End-point An endpoint is either a gateway (see below) or a terminal

Gateway There are several gateways defined to other networks (ATM, PSTN, ISDN, etc.)

Terminal User device for multimedia sessions. H.323 distinguishes between users and terminals and supports personal mobility.

MCU A Multiparty Conferencing Unit.

H.323 has been designed for local area networks and does not tell how addresses that are outside the scope of the Gatekeeper should be resolved. Terminals that registered with another Gatekeeper can only be contacted either by addressing them directly or enabling clients to ask a location server, such as an Internet Locator Service (ILS) [181].

H.323 has been criticized for being very complicated. This complexity has resulted in poor interoperability, because implementing the whole standard is a near infeasible task. Even more importantly, the Gatekeeper, by the role that it has in an H.323 network, demands to be in control of the sessions. Thus the Gatekeeper is not only aware of the sessions that it set up between end-points, but it is also responsible for negotiating the parameters of the on-going session. The Gatekeeper is thus an unwanted entity as it limits the services to the set that it is aware of (e.g., supplementary services). It thereby re-implements the call model, and severely restricts the services that can be deployed.
3.8.2 Call Management Agents

The IMTC Voice over IP Forum has proposed a Service Interoperability Implementation Agreement [176] (IA) comprising so-called Call Management Agents (CMAs). The CMAs reside on (and are associated with) CMA servers (CMAS) and are accessed by clients (CMACs by means of a CMA protocol (CMAP)). The CMACs are preferably co-located with a VoIP-terminal in the user’s device. The purpose of the CMAs is to encapsulate service behavior for a user and relate it to a set of communication terminals and addresses in different networks that are associated with the user. The specification of service behavior specifications is outside the scope of the CMA architecture, but is addressed by for instance the Call Processing Language (CPL) - see subsection 3.8.5.

The IA recognizes the fact that agents are important enablers for delivering multimedia services on Internet. However, by associating these agents with CMA servers and user addressing, its architecture severely limits user mobility, it requires the network to be available to work, and even more importantly restricts the services that can be built. Neither CPL in combination with either SIP (described below) or H.323 nor CMAs can therefore be regarded as satisfactory solutions for supporting the vision that was presented in Section 1.2.2 on page 3.

3.8.3 IETF’s Session Initiation Protocol

The Session Initiation Protocol (SIP) [59] is increasingly seen as an alternative for establishing multimedia sessions, which has been adopted as an RFC by the IETF. SIP was invented (based on research which predates H.323 — see Section 1.1.2) to make the initiation of multimedia sessions simple, light-weight, and allow (as opposed to H.323) any topology of sessions (not just calls), while clients are given the responsibility to negotiate the session parameters. SIP removes the need for the H.323 Gatekeeper and its signaling, but left it open to implementers to use H.323 channels for communication. This way, we have the best of both worlds, and SIP has quickly gained enormous popularity and support.

SIP includes the following entities:

- **Client** Software for multimedia sessions that allows a user to register with a SIP URL; sends and accepts invitations.
- **Proxy** A proxy resolves the external SIP URL, by asking a location server for the internal address (which may be redirected).
- **Redirect Server** The redirect server resolves the SIP URL by either returning the network address of the user or a redirection to another SIP Server to which the user has moved.
- **Location Server** An external entity with the ability to return the current network address of the user. This entity is not further defined in SIP.

The signaling in SIP is similar to that of other Internet protocols like HTTP, in that it uses simple
Chapter 3 Related Work

text-based primitives and arguments. Basically party A sends an invitation to party B, who responds with an acknowledgement, following this A sends an OK and the session commences. Clients are allowed to initiate multiple sessions with any party.

SIP is a UDP based protocol that facilitates the set up of multi-party sessions between multimedia producers and consumers in an arbitrary topology. There are two types of network entities involved; each with a slightly different role regarding the registration and location of users. A SIP-proxy locates a user using a location server, relieving the inviting party of knowing the current address of the invited user. A SIP redirect server will either locally resolve the address of the invited user or send the invitation on to the SIP-server where the user has moved. After the set-up of the session SIP leaves the negotiation of the ongoing communication to the end-points (using the session description protocol - SDP [60]). SIP's approach to session invocation lets the end-points do the negotiation, thus the users may invoke any type of communication in any given topology.

3.8.4 IETF’s Session Description Protocol

SDPng [94] is the short name for a successor of the Session Description Protocol (SDP) [60] developed by the IETF MMUSIC Working Group. While SDP originally was designed for announcements of Mbone multimedia conferences, new uses e.g. making IP-telephone calls imposed new requirements, spawning numerous work-arounds, sometimes heavily bending SDP syntax and semantics. This has led to a proposal for a successor to SDP (SDPng) which can accommodate those new requirements and thus address the perceived shortcomings of SDP. SDPng is targeted at session description and capability negotiation, but with a more flexible syntax, and at present not intended for supporting more complex negotiations which is one of the objectives of this dissertation.

3.8.5 IETF’s Call Processing Language

The Call Processing Language (CPL) [100] is a language that can be used to describe and control Internet telephony services (VoIP). It is not tied to any particular signaling architecture or protocol. It is anticipated that CPL will be used with both SIP and H.323 [59,178], and as such, constitutes a service architecture. CPL is based on XML and is limited in power (e.g., it provides no way to write a loop or a function) so that it can run safely in Internet telephony servers, which can validate the script prior to execution. CPL has a number of switches which allows its execution to branch depending on such things as time, location, and/or string matching. The CPL service architecture has no concept that encapsulates a user and would restrict the relation between the user and service scripts belonging to this user. Such a binding is outside of the scope of both CPL and the multimedia session controlling protocols that are intended to use it (e.g., SIP and H.323).

3.8.6 JAIN

JAIN [143] consists of a set of Java APIs to communication resources, for creating services across Public Switched Telephone Network (PSTN), packet (e.g. Internet Protocol (IP) or Asynchronous Transfer Mode (ATM)), and wireless networks, enabling the integration of Internet (IP) and Intelligent Network (IN) protocols. It is obvious from this characterization that the JAIN architecture moves back the point of service integration to the service platform, as is the case for e.g., Parlay (see paragraph 3.6.4)

3.8.7 SIPlets

SIPlets [98] or a related technique SIP-CGI, both covered in [47] offer server-side service-logic. Therefore neither of them can be a candidate for the open service architecture, which I desire.

3.9 Open Communication on Internet

This section describes how to discover services, make queries, and manipulate the results. For
dynamic events, (such as sessions) and static resources (such as printers) different mechanisms have to be used.

3.9.1 CORBA

CORBA [151] is the acronym for Common Object Request Broker Architecture. It has been proposed by the OMG [185] as an open, vendor-independent architecture and infrastructure for distributed computer applications. Using OMG’s standard protocol Internet Inter-ORB Protocol (IIOP), CORBA’s architecture is Object Oriented and built around three key building blocks:

1. An Interface Definition Language (IDL) facilitates an operating system- and language-independent packetization of software objects.
2. The Object Request Broker (ORB) facilitates the registration, localization, and invocation of software objects upon requests from clients through an IDL-stub.
3. A standard protocol IIOP enables CORBA-based programs to be interoperable.

IIOP standardizes the information, which clients and server objects can exchange and is not extensible. Furthermore, the concept of an ORB turns it into a client-server architecture, which does not scale well, when we wish to move entities out of the network and allow ad hoc communication on a peer-to-peer basis.

3.9.2 Service Location Protocol

The Service Location Protocol (SLP) [57] is a new IETF standards-track protocol designed to simplify the discovery and use of network resources such as printers, Web servers, video cameras, etc. It is positioned for client-server applications and establishing connections between network peers that offer or consume generic services, by representing resources by so-called agents:

- User Agents which acquire service handles for user applications
- Service Agents which advertise service handles
- Directory Agents which collect pointers to services

Applications running on a computer are represented by a user agent, which understands the service and resource needs of the application. A network service is represented by a service agent, which makes it available to user agents. SLP proxies are defined to act as a service agent for non-SLP applications or services. For our purposes, SLP using a directory for services does not scale well beyond the scope of client-server applications on Intranets for which it originally was designed.

3.9.3 Plug & Play Architectures for Pervasive Computing

Universal Plug & Play (UPnP) [209] and JINI [195] enable devices to connect to and use each other’s services dynamically. JINI uses a server, whereas UPnP relies upon a control point that in principle can be co-located with the resource it represents (e.g. a printer). Resources and profiles are registered during a discovery phase, then the server/control point assists in connecting devices during the lookup phase. Furthermore, the server/control point provides event services to send notifications of changes, e.g. when a resource leaves. Registrations are time-limited.

3.9.3.1 JINI

JINI is the name for a distributed computing environment that offers network “plug and play”. A device or a software service can be connected to a network and announce its presence. Clients that wish to use such a service can then locate it and call it to perform tasks. This can be used for mobile computing tasks where a service is only connected to a network for a short time, but more generally can be used in any network where there is some degree of change. There are a large number of scenarios where this can be used:
A new printer connected to the network can announce its presence and capabilities. A client can then use this printer without having to be specially configured to do so. Services can announce changes of state, such as a printer running out of paper. Listeners, typically of an administrative nature, can watch for these and flag them for attention or invoke other services.

In a running JINI system, there are three main players. There is a service, such as printing. There is a client, which would like to make use of this service. Thirdly, there is a lookup service (service locator), which acts as a broker/trader/locator between services and clients. There is an additional component, and that is a network connecting all three of these, and this network will generally be running TCP/IP.

### 3.9.3.2 UPnP

UPnP offers capabilities to support pervasive computing applications. Initially promoted by Microsoft it comprised a set of design rules that developers of software for use with Microsoft’s Operating Systems should adhere to. Currently an independent forum drives UPnP [209]. The specification is open to adoption by other operating systems, but the support in Microsoft products and UPnP’s affiliation with Microsoft are likely to encourage developers to look for solutions that involve JINI. UPnP makes use of SOAP [215] for sending control messages, and accessing objects, the General Event Notification Architecture (GENA) for events and the Simple Service Discovery Protocol (SSDP) for service discovery. UPnP does not scale well for use in large-scale mobile networks, as there is no stringent mapping between control-points, the service, or the entity that has this service. Furthermore, unlike SIP that is used in the architecture described later in the dissertation, UPnP relies heavily on the use of DNS, which raises doubts whether it will scale when thousands of mobile devices move about in the network.

### 3.9.3.3 SOAP

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols; however, the only bindings currently defined describe how to use SOAP in combination with HTTP and HTTP Extension Framework [115].

### 3.9.4 Microsoft ‘.NET’

Microsoft’s ‘.NET’ is a platform for XML Web services, which are intended to enable applications to communicate and share data over the Internet (using XML and SOAP messages) regardless of operating system or programming language. In terms of a service architecture, the ‘.NET’ platform is planned to offer a core set of building block services, based on XML, SOAP, and UDDI (see below) in addition to other components, such as authentication (Microsoft passports).

![Figure 3-11. UPnP Protocol Stack](image)

<table>
<thead>
<tr>
<th>SSDP</th>
<th>GENA</th>
<th>SOAP</th>
<th>HTTP</th>
<th>HTTPU/MU</th>
<th>TCP</th>
<th>UDP</th>
<th>IP</th>
</tr>
</thead>
</table>

Service Discovery, Eventing, Object Access
Addressing, Naming & Localization (DNS)
In summary, the ‘.NET’ platform offers an infrastructure of Microsoft products partly based on open (W3C) technology (XML, SOAP, and UDDI, which are presented in the next section). This architecture platform is well positioned to “virtualize” Windows® to become the foundation for Web services, but does not add anything in terms of support for dynamic negotiation of service behavior.

### 3.9.5 The Semantic Web (W3C)

This section discusses the open technology on which ‘.NET’ is partly based, but used instead for creating the so-called Semantic Web [9], which is a vision of a universe of Web objects, which can take into account the purpose of the user interaction. The components are presented below, followed by a conclusion.

#### 3.9.5.1 XML

The Extensible Markup Language is a general purpose format for expressing data and the foundation for the access and interchange of data on the Web [213]. XML as such offers neither more nor less. However, it is the starting point for derived notations for specific purposes, from directed graphs, logic expressions for ontologies as successors to KIF (see Subsection 3.11.1), to transactional objects in business-to-business applications.

#### 3.9.5.2 RDF

The Resource Description Framework [214] is derived from XML and is used to express relations between (XML) objects. Furthermore it has been applied for expressing machine-interpretable logic statements about objects and their relationships (see further Shoe, OIL, and DAML below).

#### 3.9.5.3 UDDI

The Universal Discovery Description and Integration (UDDI) specifications [208] consist of an XML schema for SOAP messages, and a description of the UDDI API specification. Together, these form a base information model and interaction framework that provides the ability to publish information about a broad array of Web Services.

#### 3.9.5.4 Knowledge Representation and Manipulation

Due to the increasing popularity of XML as a general-purpose data representation notation, new knowledge representation languages and support technologies have been created: DAML+OIL [166], and Shoe [190], adapting and extending results from previous research (KIF/KQML, see Section 3.11.1) to XML, in support of the vision of the creation of the so-called Semantic Web. See further Section 5.6 on page 88 for a discussion comparing KIF/KQML and DAML+OIL to some results of this dissertation.

#### 3.9.5.5 Conclusions concerning the Semantic Web

Similar to the approach in Microsoft ‘.NET’, the vision of the Semantic Web makes the assumption that all relevant communication is made through the Web as Web services based on e.g., XML, SOAP, and RDF, along with knowledge representation and manipulation. The vision of the Semantic Web purports to create intelligent communication between all objects that are accessible through the Web, and thereby restricts itself into speaking chiefly (or only) about objects which the user can observe or interact with, whereas the open service architecture that is presented in this dissertation extends this to general communication, thus going beyond a web model. Furthermore, a fundamental weakness shared by Microsoft’s ‘.NET’ initiative and the Semantic Web, relative to the purpose of this dissertation, is that both assume the existence of public shared a-priori knowledge of services (i.e., standardized) in order for these service architectures to become operational.
3.10 Ubiquitous Computing and Communication

Not only do we see services moving out of the network, but we also see computing capabilities being added at low-cost to an ever-increasing number of devices or artifacts. In addition, new short-range wireless communications allows these artifacts to communicate between themselves and to connect to a global network. This trend was foreseen in Mark Weiser’s landmark paper on Ubiquitous Computing, although his paper focused on computing rather than communication [154].

3.10.1 Linda (T-Spaces)

In order to provide even greater flexibility, researchers have investigated tuple-space based architectures (e.g. IBM T-Spaces [99,126]), in which devices and resources are able to store and share common application knowledge, by connecting to a tuple-space server. The advantage of this architecture over JINI and UPnP systems is that we can more easily add shared knowledge. In all three cases, we encounter potential scaling problems, as JINI, UPnP, and tuple-space architectures require the assistance of a server. Although in the case of UPnP, so-called control points could in principle be co-located with the agent in the device.

3.10.2 Context- & location aware computing

Context- and location-aware computing refers to the ability of user applications to take into account the user’s communication context or location and to adapt their mode of communication and user interaction. Examples of research regarding context-aware devices are Active Badges [152] and the SmartBadge [106]. Active Badges developed at Olivetti Research Laboratory (ORL) [136] facilitated the development of a “telephone receptionist aid” that gave advice on how to react to telephone calls in an optimal way, based on user context information. A second example is the SmartBadge [106], which could not only function as an ID but also includes numerous sensors and substantial computing capability. Research focusing on wearable computing user interfaces (e.g. wearables) allows users to interact with information and media in a flexible and intelligent way, examples of which are the “Remembrance Agent” [134] and “Nomadic Radio” [135]. Examples of context-aware information systems are the “Forget-me-not” [95] and the Stick-e system [96]. The former provides a human memory prosthesis, which shows relevant slices of an information base that is collected by the user and tagged with contextual information. The Stick-e system [96] allowed users to attach context-carrying information to the environment and respond to it in a relevant way.

3.10.3 Agents Overview

The paradigm of agent programming has come about as a result of research in artificial intelligence with the objective of finding solutions for creating intelligent systems with software entities that have the ability to negotiate knowledge and do reasoning in order to perform a certain task. The Foundation for Intelligent Physical Agents (FIPA) [168] has developed several standards for knowledge communication, such as Knowledge Interchange Format (KIF) [46] and KQML [180]. Programming languages and support have been developed that support the transport of functional components of these agents, or entire agents, an early example was Telescript [187], while Voyager [211] is a recent example that is an extension of Java.

The reasoning that takes place inside agents is left to the programmer and as such this paradigm is behavioristic in its approach and does not necessarily deal with cognitive processes inside the agents.

3.10.4 Mobile Agents

Research in this direction has proposed so-called (mobile) agents, agent communication languages, and ontologies (which list and specify what concepts can be discussed between different agents). The implications of these technologies for this research are discussed briefly in this section.

An agent in artificial intelligence is defined as a computing object (hardware and/or software) such
that it:

1. has a degree of autonomy in determining its behavior,
2. interacts with humans and or other agents,
3. perceives the environment and reacts to it, and
4. exhibits goal directed behavior.

Mobility stems from the software platform on which these ‘agents’ execute, which allows them to transfer all of their state, code, and execution (or subsets of themselves). Most modern systems implement this approach by adding Java class libraries to the Java Virtual Machine (e.g., Voyager [211], Aglets [97], and D’Agents [53]). Obviously, by transferring its state, code, and/or execution, an agent can accomplish tasks by gaining local control of resources it otherwise had to remotely access, or was even unable to do remotely. For instance, this approach overcomes the necessity to burden servers in client-servers systems with complexity of dealing with an increasing variety of clients; which is a problem we recognize from Intelligent Networks, where service growth has been very slow due to this problem, and due to the limitations that are associated with using access protocols that also specifies the service. Consequently, mobile agents (together with SIP) constitute the basis for our solutions, as they allow us to move the point of service integration to the end-devices.

3.10.5 Mobile Code

Mobile code is about moving the code with or without the state of execution. Different models are used in for instance: Voyager [211] and JATLite\(^2\), or Telescript. Mobile Code in itself is only a technique (see further [125] for a more extensive overview of the area), but when applied to the service architecture as presented in later chapters in this dissertation to transfer MSK and the state of execution it is an important enabler for the creation of dynamic and mobile applications.

Mobile Code is an important enabler. As we shall see in later chapters the service architecture and its components, rely on the properties of mobile code in order to create dynamic applications.

3.11 Knowledge Representation and Manipulation

In this section we examine some important contributions to the field of Knowledge Representation and Manipulation. This field of research and its applications is important from the perspective that adaptation of resources and communication requires an ability to mirror the external world; hence, the components need to carry and manipulate representations of the objects in the real world as it is observed and their relationships. The catalogue of objects in the universe of discourse is called an ontology. The next subsection mentions some notations and support technologies for manipulation of ontologies.

When we have access to such representations logic inference enables us to deduce facts from this knowledge and also infer new knowledge, which may even be represented again as logic expressions and added to the existing body of knowledge.

3.11.1 KIF and KQML

The Knowledge Interchange Format (KIF) [46] was an early attempt by FIPA [168] to capture interchangeable knowledge on a global scale, for use in communication between Agents. The Knowledge Query and Manipulation Language (KQML) [180] was created in order to enable Agents to access knowledge in other agents, transfer this knowledge between agents, or send messages (so-called performatives) instructing Agents to perform certain tasks.

The generic expression for KQML that is used has the format:

---

2. JATLite is a set of Java packages from Stanford University for writing software “agents” that communicate robustly over the Internet, by exchanging KQML messages (http://java.stanford.edu).
Conceptually, a KQML message consists of a performative, its associated arguments which include the real content of the message, and a set of optional arguments which describe the content and perhaps the sender and receiver. For example, a message representing a query about the presence of a family member at home might be encoded as:

```plaintext
(ask-one
 :content (PRESENCE Family ?alice)
 :receiver home-locator-server
 :language OBJECTS
 :ontology MESSAGING)
```

In this message, the KQML performative is `ask-one`, returning one answer. With `ask-all` we would get back an answer listing all family members that are presently home. The home-locator-server would have used performatives `tell (one)` and `tell (all)` respectively. The following is a list of performatives, which are used to assert new logical statements and query it for the truth of statements:

<table>
<thead>
<tr>
<th>Direction</th>
<th>Performative</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>ask-all</td>
<td>Query for all instances, if any</td>
</tr>
<tr>
<td>out</td>
<td>tell</td>
<td>Answer all</td>
</tr>
<tr>
<td>in</td>
<td>ask-one</td>
<td>Query for one instance, truth test</td>
</tr>
<tr>
<td>out</td>
<td>tell</td>
<td>Answer one</td>
</tr>
<tr>
<td>in</td>
<td>tell</td>
<td>Add this clause to the Knowledge Base</td>
</tr>
<tr>
<td>in</td>
<td>untell</td>
<td>Remove this instance from the Knowledge Base</td>
</tr>
</tbody>
</table>

### 3.11.2 Knowledge Maintenance

Previous attempts to create a global world of intelligent objects (e.g., an Agent Society [58]) led to interoperability questions necessitating the standardization of objects and methods in the universe of discourse (as is the case with the approach taken by FIPA [168]). Similarly ontology maintenance on the Semantic Web will run into the same problem as the approach presupposes a global common standardized vocabulary (ontology) of objects and types.

#### 3.11.2.1 Machine Learning

Machine Learning requires meta-knowledge in order to add knowledge to the Active Context Memory based on external events. There are various methods that need to be considered. For instance, genetic programming algorithms select new generations of states, based on evaluation criteria [1,92].

#### 3.11.2.2 Non-monotonic Reasoning

In an ideal world, we would like to apply monotonic reasoning, meaning that new knowledge is inferred from previous knowledge and new facts or statements (e.g. first-order predicate logic). In such cases, the maintenance of knowledge is straightforward, and allows us to identify and remove obsolete knowledge. Machine Learning contradicts this model by asserting an unproven but very likely statement based on our experiences. This non-monotonic reasoning calls for a strategy to manage knowledge (see Chapter 6).
3.11.2.3 Temporal Logic

Temporal Logic [89,118,147] is desirable to deal with the relations between time-ordered events (e.g., a certain user A promised to send a message to user B as soon as possible). Temporal logic, has been used extensively for the formal specification, verification, and simulation of distributed systems and functions, mainly in two styles of specifications, pure logical and procedural as the logical specifications can be translated to the procedural style, e.g., as in LOTOS [104]. The contribution of temporal logic is that it enables us to make statements about the time-ordering of events and states in a distributed (real-time) systems. [145] provides an example of using LOTOS for modelling and examining the properties of a GSM system.

3.11.3 RKRL

J. Mitola’s dissertation on Cognitive Radio [111] describes a Radio Knowledge Representation Language (RKRL). The purpose of which is to design “smart” radio systems based on software programmable radios that “employ model-based reasoning about its environment, location, radio propagation, networks, protocols, user, and its own internal structure” [111, subsection 4.4.9].

While RKRL is specifically targeted at creating “smart” programmable radios (cognitive radio), the knowledge representation that is required here should primarily be focused on empowering users (as well as other entities). These entities should not only be “smart” about how to react to a changing communication context, but also augment their perception and understanding of the communication context and occurring events.

Therefore, there is a different emphasis in Mobile Service Knowledge (see chapter 6), on the negotiation (through communication) between these entities and thus greater awareness of what can be known about other entities (through this communication), to satisfy the requirements, which have been listed in Section 1.3 and 1.4.

3.12 Feature Interaction

This section addresses related work concerning feature interaction, where [159] states that the problem of feature interaction stems from its own definition, indicating that defining service behavior as partial ordered sets of quanta of functionality, requires this ordering to be exact, in order to obtain the desired behavior.

In sections 3.1 through 3.6, I presented the service architectures for fixed and mobile telecommunications, as well as the Internet. These have in common the problem of feature interaction. In [159] a feature is described as an optional unit or increment of functionality and feature interaction is characterized as when a feature or features modify or influence another feature in defining overall system behavior, specifically when the behavior of one service is affected by another feature’s presence. Feature interaction was recognized as a problem in Intelligent Networks (Subsection 3.5) when services where defined as a composition of service independent building blocks [91]. [23] provides a characterization of the different categories of feature interaction. Several strategies for solving this problem were presented, e.g., with formal logic in order to detect feature interaction [15] and through negotiation and reasoning between agents [4,54,116]. As recognized by [101], the problem of feature interaction is not solved simply because communication is moved to the Internet.

An important conclusion with respect to feature interaction is that this problem may occur at two levels. Feature interaction will occur by definition when service behavior is organized as partial ordered sets of quanta of functionality, and the ordering and extensions of these specifications will cause undesirable or unexpected results, as mentioned above, these problems can be dealt with using formal specification and verification techniques. This can be contrasted to the approach taken in [111], where the (local) goal directed behavior of the device is due to and reasoning about a model, which is synthesized as a result of observed facts, and limited a-priori knowledge about radio concepts and
their relations. Therefore, in knowledge based systems (such as RKRL) there will be emergent properties, particularly the goals, which cannot be traced to a single statement. The formal specification and verification techniques that were mentioned above are bound to have limited success as they inspect only the structure of the specifications. Alternatively, with regard to the objectives of our communication, early results from attempts to resolve feature interaction in telecom services through negotiation and reasoning between agents (added to Intelligent Network service nodes) [4,54,116] are encouraging, as they address the semantics of the communication which the nodes intend negotiate.

In addition, as the Internet access allows signaling and multiple services simultaneously between end-points, the results from [4,54,116] can now be extended for negotiations, not only locally or only assisted by a third party, but directly between the communicating end-points [116]. These new opportunities are further explored in the dissertation (chapters 4 through 7) and the contributions with respect to the problem of feature interaction is analyzed in Subsection 8.3.2.

3.13 Chapter summary and conclusions

This chapter examined the limitations in existing telecom service architectures with respect to the objectives for our communication, followed by an analysis of similar limitations in the service architectures and components for second generation mobile networks and beyond. This chapter then examined novel service architectures for open communication on the Internet and ubiquitous computing, and finally analyzed to what extent languages, modeling and reasoning support for building knowledge based systems have contributed in capturing towards adding flexible and extensible service behavior in systems, in which feature interaction problems appear at different levels.

With respect to the use of agents, we found that there are scalability problems inherent to the general-purpose approach of an agent-society [58] as proposed by FIPA [168]. It presupposes the definition of concepts in universal ontologies in order for agents that were designed in different places to be compatible. Thus, FIPA has put a huge effort in designing a knowledge interchange format (KIF) [46]. But requiring the universal specification and adoption of concepts also defeats independent development of compatible service components. Clearly, this approach is not satisfactory, as FIPA does too both much and too little at the same time, and forbids us from creating usable services in different places and being assured that it will be interoperable with other service components that have been developed elsewhere without prior standardization of the content of the communication between these agents. Thus we need new mechanisms or protocols that enable the independent development of service components that are able to interpret descriptive meta-data or other service components, and collect this data for later use and infer its own reasonable response from the collected database.

What we are striving for is an open approach, which scales to the size of modern mobile networks while retaining the flexibility of tuple-spaces for coordination of entities [21] in order to be able to adapt our communication; and that includes real-time communication, which is not addressed by the World-Wide Web Consortium (W3C). JINI and Tuple -Spaces do not scale as they are server centric. UPnP protocols allow locating a control point per device, but services register with and send events to any control point that is listening. Consequently, control points cannot be regarded as representing a device, whereas we are looking for an entity that we can co-locate with any object and allow it to be the object’s representative.

Chapter 4 and particularly Section 4.6 introduce a novel service architecture that combines plug & play capabilities with support for adapting the communication to arbitrary events, and which scales well.
This chapter outlines a model and describes a service architecture for how we could and should make our communication more intelligent and adaptive. The model advocates a paradigm shift of personal communication towards ‘communication spaces’ as a way forward and as a guiding vision for this new approach and its architecture. Finally, an overview of the architecture discussing its main properties, system- and network levels, is followed by a discussion of its design and its detailed properties.

4.1 Terminology

Address: logical address of the entity according to the chosen addressing schema. This address is directly translatable to an IP-address. In SIP, universal resource locators (URLs) are used.

Agent: An autonomous, semi-permanent, software object\(^1\) that adapts and reacts to sensory data, is able to exchange knowledge with other agents, and can be mobile.

Endpoint: A communication resource is represented by an address and where applicable by an agent, e.g. a voice-gateway, a location server, or a communication terminal. An endpoint can be contacted for communication sessions.

Gateway: A media gateway between the IP-network and the switched circuit network, that provides real-time two-way communication between terminals on the circuit-switched network (commonly POTS-users) and communication terminals on the IP-network.

Location Server: Resolves the logical address of a user. In SIP this could be an LDAP-server or rwhois (RFC 2167).

4.2 Enabling Technologies

Prior to discussing the guiding vision followed by a discussion of the properties and layers of the service architecture, it is appropriate to restate a few observations regarding the role of key technologies and in what way they enable this new approach.

---

1. According to the Foundation of Intelligent Physical Agents (FIPA) [168], the definition of an agent is “an entity that resides in environments where it interprets ‘sensor’ data that reflect events in the environment and executes ‘motor’ commands that produce effects in the environment”. Such an agent has one or several attributes: autonomy, social ability, reactivity, pro-activeness, mobility, temporal continuity, and adaptivity.
4.2.1 Voice over IP (VoIP)

IP allows for multiple sessions and events over a single access channel. VoIP (exemplified by SIP-sessions) allows us to move the point of integration and coordination of multimedia communication out of the network and all the way to the user interface.

4.2.2 SIP

SIP (is an example of a protocol that) allows for ad hoc invitations to participate in multi-party multimedia communication. Even more importantly, it allows us to separate the agents that encapsulate our service behavior and the addressing & localization support. Hence, SIP enables us to move the point of multimedia service integration out to the end-devices. SIP removes the requirement that a coordinating network entity assists the communicating parties during on-going communication in negotiating (potentially varying) communication parameters. Therefore end-points and users can now be in control of the data exchanged during a session.

4.2.3 Mobile Agents

The use of mobile agents enable us to capture the functionality necessary to make the necessary decisions about communication in the end-device, this requires a certain degree of autonomy in determining its behavior when interacting with humans and/or other agents, perceiving events in the environment and reacting to these events by exhibiting goal directed behavior.

4.3 Guiding Vision: The Mobile Interactive Space

Following these observations, I formulate the guiding vision for my solution: While previously communication services were designed to center users around a network (i.e., network centric communication), we are now looking at services that allow users to conduct their business, ensured there is a network that connects them and all the things they use - i.e., user-centric computing & communication. Therefore, (assuming all the mechanisms are in place), we can view this as the disappearance of the network with the objects in the real world simply immersed in an (abstract) communication space that is overlaid on top of the real world. The Internet constitutes one realization of this envisioned communication space. The network offers uniform connectivity, naming and localization, and (wireless) mobility between entities. Residing in this communication space I propose the concept of a Mobile Interactive Space (MIS) [77,78] in which users, virtual objects, and artifacts can connect, move, and interact via representations of themselves — see Figure 4-1.

![Figure 4-1. Mobile Interactive Space](image-url)
The service architecture model which is presented in the next section constitutes a model for the realization of a Mobile Interactive Space. This model (the Mobile Interactive Space) is central to this dissertation and poses the question of how services should be provided and managed. We have already seen the implications for network access, as well as service architectures from the perspective of the limiting factors of the switched networks. Below we identify the necessary beneficial properties and requirements of the new computing and communication solutions with regard to a service architecture that enables users, devices, and virtual (intelligent) objects to participate in a mobile interactive space:

Therefore I propose such a mobile interactive space as the new service architecture model. This model is made possible by the envisioned pervasive (optionally wireless) connectivity between entities in the envisioned communication space. A computing entity in this network may represent a person, a thing (artifact), or a virtual object. Thus, people, artifacts, and virtual objects are immersed as it were in a communication space. We speak of mixed reality [67] when we design information and media such that it can be mapped on to physical objects, and conversely when changes in the physical properties or state can cause changes in the information or media. The entities in this communication space (shown in Figure 4-1) are characterized as follows:

**People**
Users of this communication space are able to stay connected (wirelessly). A personal communication identity is available that represents them and is able to act as a proxy in that it can respond to changes in the communication environment and adapt its modes of communication, as well as the communication itself.

**(Mobile) Artifacts**
Physical networked objects such as cameras, scanners, printers, robots, etc. can be represented by a similar communication identity and thus can react intelligently to events in the communication space.

**Virtual (Intelligent) Objects**
These are resources in the network that do not have a counterpart in the physical world. These resources can be made to respond intelligently to events in the communication space and act as resources. For instance, sensor fusion of information sources [29] could result in alerting a user of a certain fact.

The remainder of this chapter presents an overview of the service architecture framework and its components [78,82,84], which together constitutes a realization of the concept (and vision) of a Mobile Interactive Space (MIS) [77,78]. The concept of a Mobile Interactive Space is given a precise definition in the design of the architecture, specifically in Chapter 5 and Chapter 7; its application later is exemplified in Chapter 9.

### 4.4 Overview of the Service Architecture Framework

#### 4.4.1 Entities modeled as Agents and (Software) Objects

Assuming the presence of a mechanism for the localization of objects, the next step is to examine the functionality that allows objects to represent users, devices, and resources, and logically encapsulates reasoning capabilities in these objects. I propose the use of (mobile) agents to represent their associated objects, be they resources such as printers, a Voice Gateway to interconnect to the traditional telephony network, or SMS-servers. Indeed, we want the architecture to be open for all types of devices. Thus simple services can be modeled as resource agents, whose task is to mediate between all services that are incorporated in the architecture. The functionality of resources can now be encapsulated as objects with which other such objects can interact, and these agents act as a point of service integration.

The framework (Figure 4-2) has mandatory and optional components and therefore varies between

---

2. By reasoning about multiple events we can implement, as an example, a burglar alarm that recognizes family members, thereby avoiding sending false alarms.
application areas. Mandatory parts are the agents and the eXtensible Service Protocol (XSP). The Session Initiation Protocol (SIP) is used for naming, localization, and session initiation [82]. The Active Context Memory (ACM) is an add-on to the mobile agents, enabling mobile devices to adapt according to items 1-3 on page 8. The Active Context Memory (ACM) (Chapter 6) and SIP (naming, localization, and session initiation) are both optional (e.g., in the case of sensors only the mandatory parts are needed). Service components such as VoIP, Chat, MP3 streaming, or three-dimensional virtual spaces (3D) are also optional and can be incorporated by adding actions with corresponding session description parameters for invocation via SIP.

In the remainder of this section, I outline the necessary properties of the Service Architecture constituting a Communication Space and some of the key design issues, which will be addressed in greater detail later in this chapter, by discussing the different layers in the protocol stack and describing necessary and optional components and the relation between these components (see Figure 4-2).

### 4.4.2 Protocol Stack

The following overview of the protocol stack shows (in **bold italics**) the role and functionality of the components and how the eXtensible Service Protocol XSP [80,86] and Mobile Service Knowledge (MSK) [86] leverage SIP to provide capabilities for creating an extensible and adaptive environment with respect to user-centric communication, beyond what is possible in the UPnP architecture.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Layer</td>
<td>IP (and Mobile-IP): Connectivity</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>TCP, UDP: Sessions, Datagrams</td>
</tr>
<tr>
<td>Application Layer</td>
<td>SIP: Addressing, Naming &amp; Localization</td>
</tr>
<tr>
<td>Extension Layer</td>
<td><strong>XSP</strong>: Object Access, Eventing, Service Discovery, and <strong>Negotiation</strong></td>
</tr>
<tr>
<td>Knowledge Layer</td>
<td><strong>MSK</strong>: Adaptation, Mobile Awareness, and <strong>Reasoning about Context</strong></td>
</tr>
</tbody>
</table>

The protocol stack overview is shown in Figure 4-3. It has a number of functional components, which can be mapped onto specific protocols. It is important to note that the architecture is a framework, which implies that it is valid even if we replace some of its components. The actual mapping of functional components onto protocols is discussed below. In particular SIP comes into play in different areas, such as mobility, addressing, and session invocation. Each area is discussed in turn in the sections below.
4.4.3 Physical and Link Layers

The service architecture is not concerned with either the properties of the physical or the link layer as such. This said, service knowledge at higher-layers may be aggregated from information about the performance and behavior at the physical or link layer and possibly be used through general or special purpose interfaces to control the behavior at the physical or link layer in order to improve overall performance. For instance, when facing heavy packet loss over the existing wireless link, information about other networks might reveal a better route is available over another link that has better connectivity. If no such alternative network is available, then the mobile device might just send control packets to the receiving side, reducing its power while waiting for a certain period of time before trying again. If such a better route is available the mobile device can use this new route, perhaps adopting to its characteristics.

4.4.4 Network Layer

The service architecture is intended for the Internet; hence it relies on the use of the Internet Protocol (IP) for communication between hosts. In principle, IP is all that is required. However, the higher-layer service knowledge must be able to take advantage of the behavior of lower layers by changing its own behavior or making decisions for the network layer through general or special purpose interfaces. Secondly, the application layer in the mobile device needs to be aware of the fact that the user has moved to another network, similarly as described in the previous section, the mobile device may need to move itself to another network, in order to preserve certain communication. In fact there are different mechanisms at work to be able to establish communication properties. For a mobile device to be reachable is must be allowed access by an Authentication, Authorization, and Accounting (AAA) service, and granted a network address identity (NAI) which allows others to reach the device, even when it moves. Such a scheme has been described in [22] using extensions of Mobile-IP for IPv4 [121,122].

4.4.5 Transport- & Session-Layer

In addition to the issues that were addressed in the previous section (allowing the mobile device to be reached and to reach others even when the user moves) we also must be able to establish and maintain communication sessions.

4.4.5.1 Resource Reservation

Resource reservation mechanisms have been omitted from the discussion for two reasons. First, functionality, such as DiffServ [11,14] can be added without changing the architecture. Therefore, these mechanisms have been omitted from the discussion. Secondly, in various places in this
dissertation, evidence is presented that resource reservation is not necessary for VoIP in these wireless communication systems, as bandwidth is not a fundamental problem [72,74,81,85].

4.4.5.2 VoIP

SIP sets up the multimedia sessions, and in particular VoIP, using RTP and audio codecs for speech, such as G.723. For reference, H.323 specific signaling protocols are not shown in the protocol stack (Figure 4-3), as they have become superfluous in the presence of SIP.

4.4.5.3 A Global Communication Identity

A personal communication identity is a prerequisite for a person to be able to be located and use services. This personal communication identity has to be known throughout the entire communication space. Therefore, a globally known and implemented addressing scheme is required. On the Internet, we assume the use of a global communication identity for naming and localization of entities (i.e., users, devices, and virtual objects) for establishing and maintaining communication sessions between them. Thus, we require a name space and an address resolution mechanism for discovering and locating resources (printing services, public displays, loudspeakers, cameras, etc.) and virtual intelligent objects, or even users. The objects vary from being potentially intelligent (e.g. a robot), a resource (e.g. a printer or public display), or a simple service (e.g. VoIP call to existing telephone networks).

4.4.5.4 Naming & Localization

Whereas in switched telephony networks E.164 numbers are used to implement a universal personal number for locating users, I propose the use of SIP URLs\(^3\). For our purposes, the Universal Resource Identity (URI), and support for location resolution in the Session Initiation Protocol (SIP) [59] makes this a good choice to provide a global communication identity to the software objects (mobile agents). The simple reason for this is that SIP is an open standard (defined in IETF RFC [59]) and SIP URLs provide a sufficient basis for the identity we need. There are potential alternatives for global personal communication identities. We will not further examine these alternatives since interoperability with these alternatives can be accomplished though the use of gateways for transformation of names and for resolving of address information.

Each agent has a unique identifier (SIP URL), which enables it to be located and to invite other

---

\(^3\) [41] defines a mapping between them and method such that devices on the Internet can be located from telecom networks. Conversely, [124] defines a method such that applications on the Internet can invoke telecommunication resources.
agents or be invited to communication using SIP. Personal Communication Agents are attached to a SIP-URL that belongs to a person or other entity. Furthermore, the SIP URL provides service mobility because agents can move to other SIP-servers and use SIP to redirect communication to their new location - see also Section 4.9, which further examines mobility at different levels and places it in relation to support for naming, localization, and mobility at the network layer - see Subsection 4.9.4. SIP proxy and redirect-servers answer queries for registered users and reroute communication to the actual location of the user. Therefore the personal communication identity provides the user with personal mobility. Figure 4-12 shows an example of how this works. Via DNS & SIP redirection we can find the SIP proxy or redirect server that is able to resolve the URI to an IP-address of the named entity through the use of a location server [59] (e.g., LDAP, rwhois, WhoDP [112], or other (proprietary) instant messaging protocols [45]). Then via HTTP & SIP we establish a communication session with the object (hence localizing it) - e.g., VoIP, Chat, MP3, 3D, etc. - see Figure 4-2.

4.4.6 Application Layer

4.4.6.1 (Mobile) Agents

A personal mobile agent in the network that migrates to the mobile device as an alter ego and re-synchronizes with another instance of itself once we are on-line again ensures that other users are able to interact with us (via this personal agent which remained in the net and available) even when we are disconnected. This is an important constraint relaxation, as it would otherwise require us to be on-line and always active. Representation of resources need not necessarily incorporate such support for off-line behavior; it may be acceptable that some resources are not available when they are not connected. Re-(synchronization) can for instance be accommodated by using a standards protocol such as SyncML [197] - or using the eXtensible Service Protocol (XSP). While belonging to the application layer, the main purpose and role of XSP within the service architecture is best understood within the context of the Mobile Service Knowledge Layer, which is discussed in Chapter 5 XSP itself is described in further detail in Chapter 7

4.4.6.2 Session Invocation

We should be aware that either the representing object (mobile agent) or the represented object (resource) can initiate communication. Incorporating a SIP-protocol stack in the agent breaches this dichotomy; thus, the agent can always be aware of communication invitations (see Figure 4-5 and Figure 4-4). Even more importantly, as was discussed earlier in Subsection 4.9.4, I propose to use the mobile agent, representing the entity (i.e., the user, virtual object, resource, etc.) to do the binding for both SIP and Mobile-IP (i.e., to use SIP URLs that are already used for locating users, identical to the Network Address Identity, and have it resolved by either SIP server in either redirect or proxy-mode). Hence, our conclusion is that the mobile agent in our service architecture integrates the naming & localization mechanism offered by Mobile-IP and SIP regarding personal, service, and session mobility - see Section 4.9.

In Figure 4-6, an object representing the user incorporates a SIP capable client as well as reasoning capabilities (ACM) enabling the object to respond intelligently to events. There is a choice regarding who takes the initiative to start the communication, in particular for communication with parties who might not use a mobile agent for their communication. For incoming invitations from other such parties (e.g., to invoke a VoIP session), the mobile agent can easily be allowed to monitor communication, either through OS-services, incorporation of the SIP-stack (thus always being aware of invocations), or service specific APIs. Hence the mobile agent can still be aware of on-going communication irrespective of whether the underlying service of the agent has taken the initiative.
4.4.7 Extension Layer

The entities use a novel eXtensible Service Protocol (XSP) [80,86] – explained below and further described in Chapter 7 — for the exchange of information (semantic models, previously referred to as 'mobile intelligence') regarding the respective capabilities of entities, negotiation of (joint actions, and the subscription to and exchange of events. XSP may be a separate protocol, or as a working hypothesis we may assume XSP to be implemented as a SIP extension, adhering to [131,132].

Thus, the SIP User Agents will be co-located with a logically separate agent that handles XSP messages and delegates the aggregation, inference, and management of Mobile Service Knowledge (MSK) to its (local) Active Context Memory (ACM) -see Figure 4-6.

The purpose of XSP is similar to KQML [180] (i.e., to assist in negotiation of services between agents) with two improvements. First, ontologies (data dictionaries of service concepts) need to be agreed upon prior to the negotiation. This would otherwise be a serious obstacle to the large scale, deregulated deployment of services, where several operators could be involved. Secondly, it supports the extension of service capabilities of the agents based on communication with other agents, either by inheriting their capabilities or accumulating service knowledge based on past service behavior.

An additional purpose of XSP is to bring about a coordination of agents (mobile agents), with or without a Mobile Interactive Space (MIS) (cf. [21]). In the first case, agents rely on a MIS to accomplish a temporal decoupling as the MIS allows agents to post events, announce knowledge, and later retrieve facts from its associative memory [155]. In the later case, agents can only update each other’s knowledge through peer-to-peer negotiations.

Finally we can incorporate non-XSP-enabled information sources, communication resources, or sensors, through the use of XSP interface objects, where the interface objects acts as a proxy on behalf of the information source, communication resource, or sensor. Figure 4-6 shows the use of a proxy interface object (P) in order to incorporate a sensor to entity B. Interface proxies merely terminate the XSP protocol and lack an ACM. The non-XSP proprietary interface is outside of the scope of XSP. Examples of other uses of interface objects could be location servers or the interface object acting as a repository front end for foreground/background handling of MSK to be stored in an existing large database.

4.4.8 Mobile Knowledge Layer

The purpose of the Mobile Knowledge Layer is to propagate knowledge and implement adaptive behavior with respect to our mobile communication. The Mobile Knowledge Layer is a layer in our service architecture in the sense that its components rely on the eXtensible Service Protocol (XSP) for transport between the mobile agents. The Mobile Service Layer is comprised of several components that handle Mobile Service Knowledge, and use the eXtensible Service Protocol for propagating events, negotiating, and reconciliation of this knowledge. In terms of software layering, XSP and the mobile agents belong to the application layer, but architecture-wise, XSP is part of the Mobile-Knowledge Layer. XSP is chiefly intended for negotiation of services, propagation of events, and coordination of MSK between the agents. XSP enables agents to discover & register with each other, provide (implicit) subscription to events, and exchange service profiles. In addition, an Active Context Memory (ACM) can enhance the Personal Mobile Agent. The purpose of the ACM is to
accumulate Mobile Service Knowledge about the context of the user or the communication and infer appropriate responses to communication events based on this knowledge. The ACM is explained in further detail in Chapter 6, while MSK is covered in Chapter 5. The merits of the approach and a comparison with previous work are explained in the next section.

4.5 Merits of the Solution

In this section, I show the merits of my solution by contrasting it to previous work with existing and proposed (e.g. IETF, UPnP, or earlier the VoIP Consortium) service architectures. Some of these were already outlined in Chapter 3. This is followed by a discussion about what this approach has contributed beyond previous work.

4.5.1 Removing previous limitations while retaining scalability

The Service Architecture scales well as it does not rely on the existence of servers. Section 7.2.3 gives a detailed account of event routing and scalability. Furthermore, it takes full advantage of the fact that (due to what was said in relation to the enabling technologies mentioned in section 5.1) we are now able to move the point of integration and coordination of multimedia communication out of the network and all the way to the user interface. We are hence able to integrate any type of event and coordinate multiple simultaneous sessions. The important consequence is that we do not require the assistance of network entities or a network service (such as those offered by the network access method, e.g., POTS, GSM, etc.) to arrange the communication and make decisions for us (e.g., IN, WAP, etc.). Hence, there is no intrinsic upper-limit on the scale of services that we can build, as would otherwise be the case when all nodes that connect to the network must adhere to a fixed set of service capabilities.

In the next paragraph an account is given of how the approach taken in this chapter overcomes limitations in related service architectures that have been proposed for the Internet. By getting rid of the limitations of architectures where service capabilities were bundled with the network access methods, we are in the position to do new things. These new capabilities are further described starting with Subsection 4.5.2.

Call Management Agents (CMAs) reside on a server and are accessed by a Call Management Agent Client (CMAC), using the Call Management Agent Protocol (CMAP) in order to associate the agents with callable objects. The CMA has built-in behavior to deal with communication requests. The fact that the CMA is associated with the address of the user and located on the Call Management Agent Server (CMAS) complicates user mobility. In contrast, our service architecture applies an external addressing schema and protocol (e.g., SIP), locating the Personal Agent/Active Context Memory Agents, without requiring assistance from such an agent server. As SIP is external to the agents, it can very easily be replaced by an addressing schema and protocol with similar capabilities (e.g. Q.931 [178]). Furthermore, Personal (Mobile) Agents and Active Contexts run entirely in the mobile device...
(although they can exploit copies of themselves, elsewhere in the network). The advantage of this later approach is that on one hand when the network is available it enables user movement in the network (both handover and roaming). On the other hand, when the network is not available, the Personal Agent/Active Context Memory can still function and may even be used to setup ad hoc sessions with local devices, e.g. using Bluetooth, IrDA, etc. Since, the CMAP protocol for accessing the agents is aware of the callable objects (services), it limits our communication in the same way that e.g., CTI, IN, or other telecom architectures do. In contrast our service architecture is open and allows us to extend the communication in order to build the applications that are described in Chapter 9

4.5.2 Adaptive Responses to Communication Events

First, by introducing a point of integration and coordination at the top of the protocol stack (i.e., the mobile agent), we are in the position to make intelligent decisions about the communication. To this end, we have introduced mechanisms for describing, negotiating, and exchanging higher-level, mobile, service knowledge, which can be stored and used for inferencing in the mobile agents. These agents constitute the point of integration and coordination. The use of mobile agents enable us to capture the necessary functionality and via XSP and inferencing on the MSK in the ACM to make the necessary decisions about communication in the end-device. The result is a certain degree of autonomy in determining the mobile agent’s behavior and interaction with humans and/or other agents. By perceiving events in the environment and reacting to them, the agent exhibits goal directed behavior. This “intelligence” enables adaptation of the communication environment in the following cases:

A change in resources For instance, a user who suddenly experiences a drop in throughput may request information about the current communication environment or the communication environment might automatically try to adapt and filter information.

Changes in types of resources For example, when a public display is detected, the personal mobile agent might ask such a public display with fixed network access to assist and display the user’s information (while the user temporarily uses it as a private display). The agent can also determine if this information should be transferred to the public display.

Resources acting in concert Multiple entities may decide to act in concert as an aggregate result of an exchange of events, for instance the (re) appearance of a resource fulfilling the requirements for a particular application.

4.5.2.1 Call Processing Language (CPL)

The Call Processing Language (CPL) that has been put forward in combination with SIP in the IETF is not sufficiently flexible. Hence, using CPL as a vehicle for services and service negotiation, in the mobile agents is not a solution. This is because the services, which can be specified in CPL, are statically programmed and the dynamics of a specification are intentionally limited. Therefore, CPL in a SIP User Agent does not provide sufficient functionality to meet our requirements. In contrast, our proposed solution is much more flexible, as shown in Chapters 5, 6, and 7.

4.5.3 Seamless Services and User Interfaces

Second, the mobile agent, being the point of integration and coordination, has access to all events and with users outside or inside its feedback loop. This can include not only audio & visual, but also sensory data. Thus, in terms of applications, on one hand multimedia communication will be seamless and event-driven (i.e., ‘calls’ have no significance). The contribution of our service architecture is to generalize the notion of a call into any event by virtue of the use of XSP. Certainly calls can be accommodated for, but there are no intrinsic properties in the service architecture that speak explic-
itly of calls. Consequently, we can develop scalable communication with intuitive user interfaces ranging from context-aware audio to full-fledged mixed-reality, teleportation, or even teleport environments to us. Sections 9.4 and beyond discuss experimental data from our prototypes and test bed.

4.5.4 Ad hoc Negotiation of Services

Third, in order to facilitate ad hoc negotiation of services between entities and aggregation of devices in order to provide services, I propose an eXtensible Service Protocol (XSP), which is open-ended in the sense that it allows other agents to retrieve descriptive meta-data (mobile service knowledge) about the services from the agent in question and using this meta-data and resolution (a reasoning process) to know how to use this agent’s services. The eXtensible Service Protocol (XSP) thereby relieves the agent from having to have a priori knowledge of the functionality of resources. It is extensible in the sense that XSP provides primitives for exchanging and negotiating Mobile Service Knowledge, and for referencing its structure, but (in contrast to previous approaches) not its content. This allows services to be added without changing the protocol. An object can in principle be interrogated (see further chapter 7) for what it is able to do, by sending a request for its service profile, this is explained in further detail in chapters 5 and 7. A client is now able to discover and locate resources and interrogate them for available services. This has important implications for our service architecture, as mobile clients can now be agnostic about, for instance, how to make telephony calls, other than needing to know where to direct such a request.

Regarding optimizations: to minimize delays, in most cases we could make the mobile agent aware of common services, this enables them to locate and redirect communication to the resource that is able to handle the content (such as print files, transfer e-mail via SMS, or output it via text-to-speech conversion). But, in addition, we want the mobile agents to be able to locate, exchange, or download new functionality in order to handle new resources.

Clearly other service architectures make attempts at accomplishing this, but fail in some or most aspects. JINI allows registration of service capabilities and negotiation of services, but only using a naive type of pattern matching for finding matching services. Furthermore it is relies on servers for registration and discovery of services; hence, it does not scale.

UPnP offers negotiation of services based on predetermined and pre-agreed service profiles; furthermore although control-points can be co-located with the services that they control (e.g., an intelligent refrigerator), there is nothing inherent in the architecture that maps one onto the other.

4.5.4.1 Bootstrapping the Integration of Resources

The integration of multimedia is achieved by modeling basic services as resources, which are made available in the general model by agents that speak XSP. Thus if a user A needs to be able to contact another user B and A does not know about services that are offered by the resource R, then B can send A a portion of its XSP profile that has a reference to the location of the resource agent R (a URL). With this extension of its mobile service knowledge A is now able to use resource R in its communication with B. In case this mode of communication cannot apply to A because of the restricted communication capabilities of the device that A uses, another mode of communication needs to be negotiated between A and B, utilizing other resources. Ultimately, either A and B manage to find a mode of communication which both can manage or this fails. The resource agent (of the resource that will be involved in the communication) has a (bootstrap) method for downloading the necessary local functionality, as well as the descriptive information, which should be added to the XSP profile in A, which may now have new capabilities.

In the case of voice messaging, the Voice Server has a Voice Agent with an XSP interface and thus can be contacted by A when A needs to learn how to use voice messaging for communication with B. A already knows how to contact the agent but via XSP it now has the methods to be able to use this service.
4.5.5 Intelligent Responses to a Communication Event

The Active Context Memory (ACM) that can be associated with a mobile agent enables the mobile agent to incorporate any level of machine intelligence to inference and act on the events and mobile service knowledge MSK that are exchanged and aggregated in the ACM via XSP. The ACM is an optional component (and not a requirement) since simple entities such as individual sensors will not necessarily benefit from having an ACM attached to it. The ACM has reasoning mechanisms to infer communication decisions from this knowledge, thus implementing the intelligent behavior of our communication for purposes of augmented memory and (ad hoc) service instantiation; thus providing both. Hence, the ACM enables entities to act intelligently upon (any) events sent to this global communication identity and is thus able to deal with changes in a highly dynamic and even intelligent communication environment. Machine Learning mechanisms allow these entities to learn from our responses to communication events and subsequently act accordingly. Slices of this memory with responses related to external communication objects may be exchanged with other entities and added to their knowledge. This is reminiscent of the approach with cognition cycles taken in J. Mitola’s dissertation on “Cognitive Radio” [111], where this architecture defines a framework for the communication and negotiation between any such cognitive entities. In Chapter 5 my contributions and the issues regarding using Active Contexts are presented in greater detail.

4.5.6 Conclusions regarding the merits of my solution

The merits of this new approach are that it removes the limitations, which prevented us from creating scalable multimedia communication in mobile networks, which would otherwise require a single operator to be in control of both end-devices, access protocols, network resources, and applications. This approach leverages the inherent capabilities of IP to transport multiple services and signaling, enabling us to move the point of service integration out of the network. This further enables new modes of user communication, which can only be created if end-devices are capable of coordinating and adapting sessions to the user’s context or communication conditions. This open approach enables any party to deliver and make use of other parties’ services and applications. It is important to note that no central management is necessary other than (1) the allocation of global identities for users and resources and (2) a common language for the negotiation and exchange of information about resources.

An important contribution of this model is that it is completely independent of whether the communication agents actually encapsulates a physical person, a physical device with reasoning capabilities, or a virtual robot. These agents are able to respond to any event, be it sensory input related to the user’s context or the conditions of the communication, and infer from both these events and stored service knowledge a suitable response. In addition, by retaining and transferring this new service knowledge, other agents can benefit from this knowledge. The agents act as the point of service integration and can migrate to or be located in the end-devices. Thus, the agents constitute a means to coordinate our multimedia sessions, and adapt it to our physical, social, emotional, or communication context. Allowing agents to remain in the net and act on our behalf, should we be off-line, while learning from the agent, which went with us, causes the overall system to learn faster and be more likely to act as we wish.

Through the use of an extensible service protocol, agents can query each other and exchange meta-data descriptions of mobile service knowledge, without prior knowledge of what these services are. Thus agents have extensible service repertoires. In addition, applications can now be created through the ad hoc aggregation of multiple agents representing devices, resources, or users responding to a collection of related events (e.g., a multimedia message to a user who is in the vicinity of a networked display and loudspeaker, can itself arrange play out using these devices, if this is what the user would expect & wish). The communication agents can also represent virtual spaces, which can be mapped onto physical places and thus create mixed reality applications.

By collecting and monitoring stored service knowledge service, service providers can learn, predict,
and through simulation even forecast the success of services for a faster, more cost-effective, and more adaptive design of communication solutions.

### 4.6 The Design of the Architecture

The previous sections of this chapter described the general structure of the service architecture by explaining the purpose of and relationship between different components of the service architecture, level by level. In this section and the following ones that we will examine the design of the service architecture, by studying how the components of the service architecture co-operate in their operational environment. In order to do so, we will first study the communication environment in which the components of our service architecture operate.

### 4.7 Operational Environment

Figure 4-8 shows the major components and nodes in the operational environment for our service architecture. The overview does not show components for Mobile-IP since the scenario for and relation between SIP and Mobile-IP for different types of mobility are covered in detail in Section 4.9.

#### 4.7.1 Integration of Legacy Services

In our solution (shown in Figure 4-8) we integrate legacy services for personal communication (e.g., POTS and GSM) via switched access (indicated by the customary X-like symbol for switched networks) through a Voice Gateway with the Internet (indicated simply as providing connectivity through IP). The purpose of the Voice Gateway is to interface between voice sessions on the Internet via SIP and voice calls via switched access in telephony networks. The Voice Gateway contains a SIP-stack to handle session invitations. Optionally it is integrated with a directory, in order to be able to resolve telephone numbers to SIP URLs and vice versa for calls in both directions (see also [124]).

In Figure 4-8, we also see a Personal Agent being co-located with the Voice Gateway, running on an agent server located on another host. The Personal Agent is the master-copy of the Personal Agent that is running on the mobile, which means that it is in stand-by mode as long as the mobile is active. The (Personal) Agent Server is co-located with the Voice Gateway for practical reasons, as we may expect incoming calls from the legacy networks, when we are off-line [69] and hence is a design decision for which there are alternatives. Given [41] the Voice Gateway can determine the address of the Personal Agent, which effectively can reside anywhere on the Internet. In addition, the Personal Agent in the mobile is the active copy, which is contacted and synchronizes its data with the master copy, which is discussed in the next subsection. A SIP Server (in redirect or proxy mode) assists in the invocation of sessions between SIP User Agents (UA) running on end-points, be it a mobile device, a resource, or a software object.

![Figure 4-8. Operational Environment.](image-url)
4.7.2 (Logically) Off-line Behavior

In order to support off-line behavior, either physically or logically off-line we may design the Personal Mobile Agent such that it has a minimal set of capabilities to locate its master copy running on the agent server in its home location (signified by its SIP URL) and synchronize the state of the master copy with the one that is running on the mobile host. When going off-line the Personal Mobile Agent in the mobile device allows its data to migrate back to a master copy that is running on the agent server, and gives it the initiative to act in its place. The Personal Agent is thus able to act even when relocated to the agent server. Naturally, the SIP servers must reregister the mobile agent to have it moved back to its home location. In Figure 4-9, the basic scenario is shown: $A_{\text{off}}$ is the off-line location of A.

1. Registration
2. Lookup of A’s location
3. Session B-$A_{\text{off}}$
4. Relocation (migration of data)
5. Re-registration
6. Lookup
7. Redirection
8. Session B-A

Even more importantly, this scenario works well even in the case that the Personal Mobile Agent may loose contact (e.g., through disruptions in the wireless communication link, batteries going dead, or the user shutting down the device unintentionally). The Personal Mobile Agent replies to keep-alive messages from the Personal Agent. Depending on the application scenario, we may optimize how aggressive we should be in sending information deltas about changes of data in the Personal Mobile Agent, and in many cases simply sending keep-alive messages will suffice. For instance, in the case of just browsing through voice mail, temporarily loosing communications does not requires any updates (i.e., synchronization) at all, since any changes are saved on the voice server anyway.

4.7.3 Adding Context-Awareness

The Personal Agent that runs on the mobile device uses co-located or integrated functionality: a SIP User Agent (UA), the eXtensible Service Protocol, and the Active Context (Memory). In addition, we may integrate sensors, which through the use of a Personal Agent and the Active Context may generate events, which are forwarded to other agents (or the Location Gateway). Sensory input can be integrated in different ways:

- A sensor agent (e.g., Beacon Server [90]) may be running on the mobile device, which forwards a software signal with the URL of the detected beacon; as the communication is carried out using IP, this component can be relocated to another host. Thus the sensory input to the personal agent can “teleport” (i.e., move its locus of execution) to other devices. In Figure 4-8, this is shown as a stand-alone sensor and sensor agent. XSP facilitates negotiating of the communication.

- If a sensor is integrated in the mobile device, then it is assumed the mobile agent either has access to the sensor data via a special-purpose API or a Java-interface. This model applies to wearable computing where there is a tighter integration between the mobile device and sensors. In principle, a device’s sensors can be made available to mobile agents on other devices by applying the same method as in the previous paragraph; i.e., by adding a sensor agent and an XSP interface.
4.7.4 Management of Location Data

In order to be able to resolve the address of the end-point, SIP Servers may use a Location Server. In Figure 5-8, we use a Location Gateway (GW), in order to be able to update the information in the Location Server. Consider for instance, position data from a server running an engine with a model and software objects that have three-dimensional properties, these objects can be rendered, viewed, and visualized in clients on the mobile [20,83]. The Location Gateway can be enhanced to aggregate other position data from movements of either the user or the mobile device [7], in particular handoffs of mobile devices between WLAN access points, location changes from Mobile-IP, and detection of URLs from objects by the Personal Agent that is running in the mobile device can be subsequently translated and forwarded to the location gateway as position and location update events. To be able to manage the data for certain locations, we need to do more processing in the Location Gateway, which can be accomplished by adding a Location Agent to the Location Gateway. The Location Agent has then two tasks: first it must do sensor fusion of location updates and position data from different sources and secondly it must store the results on the servers, e.g., the Location Server (which is a database), and the 3D engine (which holds the attributes of the dynamic software objects). This allows us to correlate logical and physical positions with addresses for communication, thus enabling mixed-reality applications.

4.7.5 Human-Computer Interaction issues

In order to avoid the limitations that we noted in telecom service architectures - see Chapter 3 - we have moved the point of service integration out of the network into the mobile host. The behavior of the user interface is dictated by the behavior of the basic services, the action of the Personal Mobile Agent in response to communication event, matched to the capabilities of the mobile device. As described in Subsection 4.7.3, we may integrate sensors in our user interface, even those that are not located on our mobile device. Quite in parallel to this we can also involve communication resources that are not located on our mobile device. This means two things: in the first place we can extend the scope of our communication, interaction to other devices (even ‘teleport’ to such devices, e.g., involve a camera and haptic devices, which are remotely located) and in the second place we are in the position to dissemble our mobile device into any number of wearable items, which are able to locate each other and communicate. Probably they communicate in an ad hoc network, where one of the nodes agrees to route packets to/from external nodes (as it has sufficient battery capacity to sustain the required wireless communication). (This role of gateway to PAN to WLAN might move from device to device over time or even be shared by multiple devices.)
4.7.6 Agent Communication

The agents communicate via a common protocol that is negotiated after localization and negotiation of the communication. In earlier prototypes (see Section 9.6), KQML (a text-based transport mechanism) was used. Using KQML requires agents to incorporate capabilities not only to interpret the messages and extract the knowledge statement, but also be aware of the ontology (knowing what concepts the agents can discuss). Therefore it has been replaced by the extensible service protocol (XSP) that has been proposed in this dissertation (see Section 4.4) and explained in further detail in Chapter 7.

In Figure 4-8, an Agent Server is depicted. The Agent Server is not inherent to the solution, but was used as the result of a design decision to use and adapt the JATLite agent platform for our research (see Subsection 3.10.3). In a later redesign of the prototype we successively used software support for implementing mobile agents (e.g., Voyager [211]), and ultimately a Distributed Agent Environment for Mobile Object Networks (DAEMON) [3], in which no central Agent Server is present, it only requires the use of a class library, which enables communication via a common registry and access control agents.

4.7.7 Peer-to-Peer or Server mode

Peer-to-peer mode is the default mode for agents to operate in, whereas Server mode is simply a special case of the more general peer-to-peer mode. One of the agents can volunteer to represent an abstract space notion in which other registered agents constitute a “neighborhood”. The “server” agent then sends notifications to the other agents. Agents that operate in server mode and represent an abstract space, volunteer to explain how objects in its space are interrelated to other agents that are not part of its ‘neighborhood’.

The following sections address design decisions with respect to the service architecture or general considerations regarding use of underlying services and protocols.

4.7.8 Personal Agent

In earlier designs, we created separate agents for the user interaction (a Client Agent, as well as the Active Context Agent; note that these were design decisions and not inherent to the solution. The Personal Agent can communicate with any agent it has knowledge of, and thus our design allows us to create and use such helper agents. At a later stage we abandoned this approach, as it offered no obvious advantages with respect to the communication in our application prototypes. Should such an approach be warranted for certain applications, then the identities of the Client Agent and the Active Context Agent could be made a-priori knowledge and not need be looked up. Other resources can also be represented as well-known agents to the personal agent in the agent system and contacted via XSP (e.g. the Location Agent).

4.8 Agent Properties

In the previous section we discussed the design of the architecture. In this section we will examine the design and properties of the mobile agent that enables the ad hoc application networking (the topic of this dissertation), outline the agent’s general properties (see Figure 4-10), followed by describing the design choices regarding each property (see Table I and Figure 6-3 on page 94). As we have seen, an XSP part is needed to
communicate with other agents. The agents also must implement some meaningful behavior in response to XSP-messages or be invoked by the SIP part that handles invitation to sessions. The SIP-part is not mandatory, but then the agent is most likely a helper agent such as the location agent, as described in Section 9.6 and is used in a wider context. The agent may or may not include a graphical user interface (GUI). The agent may or may not include a Query Interface, Knowledge Base, or Inference Engine; these three comprise the Active Context Memory. Figure 6-3 shows the agent’s properties in further detail. Table I lists their properties, including if they are mandatory or optional, followed by the design choices and implementation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Required</th>
<th>Design Decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Agent Communication</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The agent must be able to communicate comprising:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. A communication identity (ID)</td>
<td>Mandatory</td>
<td>SIP URL</td>
</tr>
<tr>
<td>2. Account Data</td>
<td>Optional</td>
<td>AAA data</td>
</tr>
<tr>
<td>3. The ability to send and receive events</td>
<td>Mandatory</td>
<td>XSP</td>
</tr>
<tr>
<td>4. The ability to register subscriptions to events and route events to subscribed entities</td>
<td>Mandatory</td>
<td>XSP</td>
</tr>
<tr>
<td>The agent can negotiate its service behavior, thus it has:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. the ability to query other agents about service capabilities</td>
<td>Optional</td>
<td>XSP</td>
</tr>
<tr>
<td>2. the ability to respond to service capability requests</td>
<td>Optional</td>
<td>XSP</td>
</tr>
<tr>
<td>3. a profile with states and service capabilities</td>
<td>Optional</td>
<td>RDF [214]</td>
</tr>
<tr>
<td>The agent presupposes the existence of:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. A naming and localization schema</td>
<td>External</td>
<td>SIP</td>
</tr>
<tr>
<td>2. A mechanism for the invocation of multimedia</td>
<td>External</td>
<td>SIP</td>
</tr>
<tr>
<td>3. Mobility Support</td>
<td>External</td>
<td>Mobile-IP</td>
</tr>
<tr>
<td><strong>Agent Presentation</strong> (the Agent has a GUI)</td>
<td>Optional</td>
<td>Platform dependent and can be part of the profile.</td>
</tr>
<tr>
<td><strong>Agent Behavior</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The agent can aggregate, store, and retrieve Mobile Service Knowledge (MSK), and thus has an Active Context Memory (ACM), with the following mandatory/optional components relative to the ACM:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. A data store for MSK</td>
<td>Mandatory</td>
<td>Prolog [189]</td>
</tr>
<tr>
<td>2. A query interface for retrieving MSK</td>
<td>Mandatory</td>
<td>Jasper [179]</td>
</tr>
<tr>
<td>3. Background/foreground handling</td>
<td>Optional</td>
<td>Prolog</td>
</tr>
<tr>
<td>4. Garbage Collection</td>
<td>Optional</td>
<td>Prolog</td>
</tr>
<tr>
<td>5. Inference Engine</td>
<td>Optional</td>
<td>Prolog</td>
</tr>
</tbody>
</table>
4.8.1 Agent Profiles

Agents have an external communication profile (shown formatted as XML below) that is used in XSP and that specifies a number of properties:

```xml
<object>
  <type> // ontology
    <!-- space -->
    [// optional
      [<db>]
      [<sensor>]}
  </type>
  <actions>
    <action>
      <event> // trigger templates
        <!-- event -->
      </event>*
    </action>
  </actions>
  <states>
    <public> // visible states
      <state>
      </state>
    </public>
    <state> // invisible (private) states
    </state>
  </states>
</object>
```

4.8.2 Types

The `<type>` tag has a reference to a Resource Description Framework (RDF) file [214], which acts as the XML [213] equivalent of an ontology. Ontologies are described as relations between agents. The ontologies are stored as Resource Description Framework files. The RDF file contains resource meta-data modeled as object relationships between resources. Other agents look for semantic equivalence to one of the related resources during registration and are thus able to enter an ad hoc application network, without requiring user intervention. The RDF files are either created by a system administrator, generated by the Active Context Memory (ACM) as a result of successful use cases, or default RDF files for the following well-known types.

- `<space>`: Agents can enable a space tag (optional), indicating that it will propagate information about all registered agents to other agents. The space tag default state is that it is disabled. The enabled tag allows an agent to act as an abstract space. Other agents registering with this abstract space obtain information about other registered agents (i.e., objects that are present), and can register with these other agents. Agents with the space tag enabled will forward notifications of events that have been received from one of its registered agents to all other registered agents; while agents with a disabled space tag will not. In case of adjacent agents with a `<space>` tag, an ordering of the two is enforced according to the semantics of the relational graph in the RDF file (see also Subsection 6.3.10).

- `<db>`: Agents can also volunteer to act as a repository for background service knowledge for other agents, by enabling the (optional) db tag. In principle, agents can volunteer to act both as a space and a repository, but there is no requirement that they should. XSP offers a limited default set of actions for knowledge storage and retrieval with regular expressions. The ACM can optionally enforce more semantics to these regular expressions, relative to the MSK that is currently subject to reasoning. — see further Table I, “XSP Messages,” on page 106.

- `<sensor>`: Sensors can be monitored by agents either via an internal interface, or via an external API (e.g., Java RMI). Stimuli from the sensor cause the agent to act. Details are outside of the scope of XSP. However, by setting the sensor tags, agents will understand that this agent has only mandatory functionality, and its actions are limited to monitoring and setting properties. The sensor tag is mutually exclusive with both the db and space tags. Agents of this type allow us to build and incorporate sensor networks in our applications.
4.8.3 Actions

Actions are rules that are triggered by a logic expression of event patterns, allowing it to contain variables or not even to be fully instantiated. Actions can be represented as XML. The action repertoire can be extended as a result of internal reasoning in the ACM or more likely as the result of negotiation with another agent resulting in it sending the mobile code for this new action to add to the receiving agent’s repertoire.

Actions are triggered by events. Events may be added to the list of triggers either by the Active Context Memory or by another object using XSP.

There are general-purpose actions, which enable another agent to set these public states. Public state changes result in notifications to other objects.

4.8.4 States

States can be either public or private. Actions cause state-changes as side effects. Therefore, exporting actions also results in exporting states to the receiving agent’s profile.

4.9 Mobility

This section examines the requirements on support for mobility with respect to multimedia service delivery. More importantly, it examines how we can take advantage of the new properties and constituent components of multimedia communication now that it has been moved to the Internet, concluding that application level decisions can further relax the requirements on the network and end-points.

Multimedia applications on mobile hosts in IP-networks must be able to deal with different types of mobility. This section examines mobility scenarios in order to determine the feasibility of using mobile hosts in wireless networks to provide multimedia services over IP.

In Section 4.4, in an overview of the service architecture, we describe the location and relation between several protocols, in particular SIP and Mobile-IP. SIP and Mobile-IP are non-orthogonal regarding naming, localization, and mobility. Clearly, we need to combine SIP’s application-level naming and localization scheme with other mechanisms (e.g., Mobile-IP) in order to assure that our multimedia communication can follow (via roaming and handover) a user or device that moves (e.g., [153]). Therefore it is appropriate to describe the relation between the different components and protocols in the layers of the service architecture framework in order for them to display the intended meaningful behavior. In order to be able to do so we need definitions of mobility concepts at different levels:
We speak of personal mobility when others are able to reach us using a logical identity (e.g., URI in SIP), irrespective of what device we are using or our location in the network, or even which network we are located in. It does not speak about what others are able to do, once they have reached us, only that they can reach us.

We speak of service mobility when a service is available to us, when we have moved to another location with other but similar resources. For example, in telecom networks, we would like a wake-up call to follow us when we move from PSTN to GSM. On Internet we would like to use a print service which uses the nearest printer, not the one we left behind.

We speak of device (or network) mobility when others or other devices are able to locate a device using (one of) the network address(es) of this device.

Clearly, these concepts come together in our service architecture, and we must clarify how the different components and protocols relate in order to provide personal, service, and device mobility.

4.9.1 Personal/Entity Mobility

To support personal mobility, a person should be presented with the same services and environment irrespective of where they access the network. This is achieved by resolving the identity of the user into the address of a repository where the user’s data is stored and binds this address to his or her current communication profile. As pointed out in subsections 4.4.5 and 4.4.6, I propose to bind SIP URLs to the personal (mobile) agents, which also incorporate a SIP User Agent (UA). This UA is simply a container for the SIP functionality. As the communication profile, modeled as a (mobile) agent, represents any entity not just persons, but also (virtual) objects or resources, these objects are also reachable and mobile in the ‘personal’ sense of mobility.

4.9.2 Service Mobility

An interesting consequence of this approach is that ‘personal’ mobility enables service mobility as the (mobile) agents representing (virtual) objects or resources, may migrate to and from the user’s device, thus executing in another location relative to the current communication context; hence providing ‘service’ mobility at the same time as personal mobility.

4.9.3 Device Mobility

A handover between access-points on the same LAN-segment does not change the network address of the mobile and is handled completely by the link layer protocol (e.g. IEEE 802.11) - consequently, it causes no problems. There is only a small delay due to the link layer bridges needing to learn the MAC address and to change the spanning tree — see Section 9.4.6 on page 139.

However, when the host moves to another network segment, it receives another IP-address. Clearly, AAA functions need to be involved in this case in order to grant the user re-attachment to the network (or an event to another operator’s network). Mobile-IP [121] is needed to resolve the situation that otherwise arises when a corresponding host is given no knowledge of these movements and packets are sent to the mobile host’s old IP-address. A detailed description of Mobile-IP is not within the scope of the dissertation, but on a general level, this situation is resolved by using a so-called Home Agent (HA), the address of which is fixed. The HA is used by the Corresponding Host (CH) in order

4. “Personal mobility is the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move. Personal mobility is based on the use of a unique personal identity (i.e. ‘personal number’).” [119, p. 44].
to forward packets (via an IP in IP tunnel) to the Mobile Host (MH). Traffic is then tunneled between the HA and FA using Mobile-IP without applications needing to be aware that Mobile-IP is being used. In order to assist the MH in finding a new IP-address and de-tunneling traffic from the HA, a so-called Foreign Agent (FA) is used - perhaps co-located in the MH.

### 4.9.4 Session Mobility

There are cases of mobility involving sessions where a change to another physical location in the network requires attention, namely: handover and roaming. These cases need special attention as more than the network layer (i.e., Mobile IP) may be involved. As illustrated by the following example (see Figure 4-12), both the transport layer and the application layer may be involved.

Initially the user is using mobile host (MH\(_1\)) and registered (1) with a SIP server. The SIP server receives an invitation (2) from the corresponding host (CH) and may use a location server (not explicitly shown) to resolve the actual location of the MH\(_1\). When the user and the mobile device move (m) to another location in the network (i.e., handover, involving a change of IP-address in NW2), we may choose to hide altogether this movement from the multimedia communication with Mobile-IP (using a Home Agent) or (as shown in Figure 4-12) we can detect the movement in the User Agent (UA\(_1\)), reregister and send a new invitation, or just redirect the packets in the multimedia communication (e.g., RTP packets using the RTCP control channel) to the new address. Obviously, there are disadvantages when we lack a clear distinction between what components handle the mobility. Below I will describe a proposal, which improves the situation.

When the mobile user moves to another network (u) and re-registers there, the invitation by the corresponding host (CH) is sent to the location that was known to it earlier (5). The invitation is forwarded to the new location, where it is resolved (6). The CH is now able to establish communication with the MH\(_2\) in the new location.

### 4.9.5 Roaming

As we have seen in the scenario above, roaming need not involve complexity due to multimedia sessions that must be retained while moving the mobile device to another location. When the user (or entity) re-registers in another network, we simply need to ensure that the client on the corresponding host is able to locate the mobile user in their new location on their new host. For user mobility we propose to use SIP to resolve the actual location of the user, enabling the user to roam in the network.

---

5. The performance of handoffs for unicast and multicast were the topics of the KTH licentiate theses of J-O Vatn [148] and Jiang Wu [157] (respectively).
Note, that it is the user who moved: therefore, the user is not necessarily using the same device, as illustrated in Figure 4-12.

Recently, extensions of Mobile-IP have been specified in the IETF [22], allowing mobile hosts (MH) to register and be reached using Network Address Identifiers (NAI), similar to URLs used in SIP. In the case of movement to other networks, AAA functionality [63] will be involved to grant the mobile node access to the network through contacting the AAA Home (AAAH, i.e., DIAMETER) in order to assign a dynamic mobility anchor point (dynamic Home Agent). In the case when users are located in the same area, we avoid triangular routing, as would have been the case when the Home Agent of one of the users was located in a remote network.

We propose to use SIP servers (as SIP URLs are already used for locating users) to act as the DNS-like entity in order that Mobile-IP can use the same URL as the Network Address Identity and have it resolved by either SIP server in either redirect or proxy-mode. The bottom-line in this approach is that the mobile agent, representing the entity (i.e., the user, virtual object, resource, etc.) is able to do the binding for both SIP and Mobile-IP. Therefore, this functionality can be embedded as subclasses in the mobile agent and hidden from the user and other entities.

4.9.6 Handover

Clearly, in the case of real-time communication, in particular voice, we need to take special precautions in our Service Architecture in order for multimedia to be mobile on the Internet. When the mobile’s communication is handed over to an access point on another network segment, we need to ensure that both the signaling and session (between the corresponding host and the mobile host) survive the relocation of the mobile host in the IP network. In the next section a number of methods are mentioned that can and have been applied to our Service Architecture (Chapter 4), demonstrating the feasibility of mobile multimedia, without going into detail concerning these solutions.

4.9.7 Mobile multimedia handovers

As above, we assume device mobility is achieved through Mobile-IP. Since in the case of Mobile-IP when the mobile host moves to an access point in another network segment, it will receive a new temporary network address. If real-time isochronous communication (such as voice) is not in use, we can simply use Mobile-IP [121] to enable the corresponding host to continue to communicate with the mobile host and via its home agent remain agnostic about its actual IP-address location in the network. However, the triangular routing that is caused in Mobile-IP [121] by sending packets to the HA causes additional delay, and has led to numerous proposals in order to avoid the extra latency for real-time transport - for instance, route optimization [123] or hierarchical routing in order to be able to do fast-handoffs [24] (additional link layer mobility may be used - e.g., between WLAN access points - facilitating even faster handoffs).

Overall we are seeing a movement to an approach where the user negotiates network access (the point of attachment) such that the user is allocated a dynamic home agent, and where a DNS-like

![Figure 4-13. Mobile-IP Extensions](image-url)
entity resolves the Network Address Identifier into the location of this dynamic Home Agent. This approach is well aligned with the use of the Session Initiation Protocol for reachability and invocation of multimedia sessions, while also avoiding triangular routing (see paragraph 4.9.5).

Triangular routing in Mobile-IP occurs because of the use of a home agent (HA) to hide movements of the mobile host. Minimizing additional latency is addressed by route optimization [123], which solves this problem by sending binding updates to inform the corresponding host about the actual location (IP-address) of the mobile host. However, for real-time services, such as VoIP that use RTP streams during multimedia sessions, route optimization does not work well [153]:

Only the home agent may send binding updates to correspondent hosts. This means that there will be an extra delay before the correspondent host finds out where to send the packets, during which time the old foreign agent must forward the packets to the correct location; but the old Foreign Agent (FA) doesn’t necessarily know where the mobile is now.

The mobile host needs to rely on the old foreign agent forwarding packets to its new foreign agent until the correspondent host has got the binding update. However, there is no requirement saying that the foreign agent must do so. Wedlund and Schulzrinne [153] propose using knowledge in a higher-layer protocol SIP to be able to do fast handoffs with low latency and low loss of real-time data as a solution for the two problems mentioned above. The HA and FA are used to carry the SIP-signaling but the RTP streams related to real-time sessions are renegotiated by SIP and hence do not rely on use of the home agent and foreign agent in Mobile-IP. For SIP to be aware of the movement of the mobile host, the SIP server could for instance be co-located with the home agent in order to be able to query the HA for the current location of the mobile host. However, renegotiation of the session probably takes longer than just updating the HA; hence it is not clear that this method really is faster, only that it allows the CH to find a direct path to the mobile host for the RTP traffic.

Furthermore, there are other possible approaches for supporting multimedia mobility, such as hierarchical mobility management [28] or the use of cellular-IP in the local area for fast handoffs, while retaining Mobile-IP in the backbone [24], that also merit further study. However, these are optimizations and not critical to this dissertation. In addition, alternative strategies have been proposed for third-generation cellular networks (3G), where devices retain their IP-address even after a hand over, by delegating device mobility support to the link layer [205].

4.9.8 Remote Networks

When the remote side is located on another network (e.g., GSM or the PSTN), a gateway is used to interface sessions and/or signaling (e.g., a voice gateway [69] and PINT [124]) to act on behalf of the remote device (e.g., a mobile phone). We successfully implemented such gateways both for H.323 and SIP, gaining experiences regarding their feasibility, scalability, and use in services [69,70,71] - see Chapter 9. Voice- or rather media-gateways are now available from many vendors.

4.10 Invocation of Resources (e.g., VoIP)

Figure 4-2 on page 56 shows how the different components in the application were integrated in our service architecture. An agent represents each user, device, or service. The SIP-redirect server, Voice Server, and Voice Gateway represent the basic VoIP services that are available. Agents may use them directly by connecting via middleware, such as for instance Java-RMI [194], Voyager [211], or Daemon [3]. Another possible approach is to utilize a SIP-agent whose task it is to relieve other agents from having to have knowledge about the VoIP infrastructure.

Resources or entities that are not part of the agent system are thus contacted by means of SIP. Hence, the Personal Agent must be able to receive a SIP-invitation, with an SDP-description of what session is required, such as an invitation to play games, or use VoIP. If the calling party has the capabilities to converse in XSP, an XSP session is invoked. For example:
4.10.1 Integration of Devices

Gadget Memory [56] allowed devices to be integrated seamlessly and represented by such agents through a distributed shared-memory.

4.10.2 Example

Figure 4-14 shows an example of the communication, summarizing how the service architecture can be used: Two users (tom@fatburen.org and theo@it.kth.se) are connected through different access networks (GPRS, WLAN) to Internet. The mobile devices obtain IP-addresses from Mobile-IP assisted by AAA and the mobile agents register with a SIP server. Somehow (e.g., through visiting a web page announcing this service), the agents find out about a Bågen group server and register with this agent, which represents an abstract space. In this abstract space, there are locations, which can be visited through the personal agents generating events, for instance, by physical instances receiving stimuli from beacon sensors or real-world positions from GPS sensors, or simply browsing the user interface that the Bågen Server sends. Based on these events, a VoIP session is invoked between the two parties when a rule in ‘Bågen’ is triggered showing that ‘tom’ and ‘theo’ are sufficiently close for them to logically “hear” one another.

4.11 Concluding remarks concerning this new service architecture

This chapter has shown the foundation of this service architecture showing its operation environment and how the different components cooperate within this environment. The important contribution of this service architecture, in comparison to limitations in the ones presented in Chapter 3, is that fully leverages the fact that wireless packet network access to Internet, enables us to move the point of service integration to the application layer and out of the network to the
end-devices. The design of service architecture enables users (and other entities) to move in the network and retain their communication, and at the same time the entities which participate in the delivery of the service are enabled to make optimal decisions regarding the purpose of the communication (optimal with respect to either the user context or the communication context). Moreover, the service architecture, enables entities to discover resources and negotiate regarding the services, without requiring detailed knowledge to be shared and specified in advance, which previously limited the proliferation of agent-based services. This service architecture offers plug & play support for services, but unlike JINI does not require the assistance of a third entity and unlike UPnP has no specific mapping between control points and services. Furthermore, the service architecture leverages SIP and Mobile-IP in order to scale Mobile Interactive Spaces (cf. smart spaces) to the size of large scale mobile networks.

In the next few chapters we will study in further detail some components, which enable entities to display adaptive behavior. Chapter 5 covers the properties of Mobile Service Knowledge (MSK) and how it is interpreted & acted upon by entities, causing them to be aware of and adapt to changes in availability and types of resources. Chapter 6 examines the structure, components, external interfaces of the Active Context Memory (ACM), which holds components for storing, managing, evaluating, and acting on the Mobile Service Knowledge. In Chapter 7 we learn the principles for how the eXtensible Service Protocol facilitates the routing of events and exchange of mobile service knowledge between entities. XSP, the ACM, and MSK are all integral parts of the service architecture.

Chapter 8 discusses “Network Aspects and Requirements” regarding the consequences of the service architecture regarding provisioning of services. This is followed by a discussion of examples and prototypes, which demonstrate the feasibility of the service architecture and illustrate some new modes of user communication that have been enabled or whose design has been simplified.

Sections 9.4 and beyond of Chapter 9 provide more examples of how these results have been applied.
5 MOBILE SERVICE KNOWLEDGE (MSK)

Given the service architecture, which was presented in the previous chapter, we are now in the position to examine the properties of Mobile Service Knowledge as a vehicle for intelligent, adaptive behavior that we want our agents to display. Following some introductory remarks regarding its relation to RKRL (Subsection 3.11.3), we will discuss the knowledge representation of MSK, mechanisms for interpreting and acting on this knowledge, and finally a discussion of design decisions as well as the current implementation.

5.1 Introduction

We need to enable applications and resources to be aware not only of the configuration of their immediate environment, but also to learn how these resources are related and automatically configure themselves to achieve a collective purpose. Such machine interpretable descriptions will hereafter be referred to as Mobile Service Knowledge (MSK) [80,86]. MSK is an integral part of our service architecture and resides at the top of the protocol stack as shown in Figure 4-3.

On an abstract level, MSK is not merely a specification tool for service behavior, which interpreted by support for reasoning, causes dynamic behavior in our systems. With respect to the guiding vision for the communication, which was presented earlier in Section 4.3 on page 54, MSK is a (machine interpretable) description of the world as observed by an artifact, virtual object, or user. Thus, MSK fulfills a dual purpose of on one-hand describing the (observed) objects in the world, the relationships of the observed entities in it, including the entity that carries this MSK and on the other hand constituting the behavior of this entity — in combination with the reasoning in the Active Context Memory (ACM).

From a system point of view the combination of MSK with reasoning - supported by the Active Context Memory (ACM), see Chapter 6 — constitutes mobile intelligence. Mobile intelligence may consider the user (adaptation of content to a specific user need or in a specific context), consider the application (auto-configuration and optimization of network resources, e.g., for the opportunistic delivery of content), or consider the resources & devices (self configuration for local operation optimization, e.g., power management or link utilization). The purpose of MSK is specifically to capture knowledge and thus empower the agents of users (and thus the users themselves) to make intelligent decisions regarding use of local communication resources, modes of user operation, as well as interaction with services belonging to other entities. Therefore the knowledge representation chosen for Mobile Service Knowledge must:

- (inwardly) reflect awareness of the presence of entities, and the availability or non-availability of resources,
- (inwardly) mirror the perceived entities and the relations (categories of events) between them in partial orders, and
• (outwardly) exhibit capabilities of the services which the entity is able to offer to the user or to other entities.

This means these entities should not only be “smart” about how to react to a changing communication context, but also augment their perception and understanding of the communication context and occurring events. They should automatically volunteer to offer their new-won capabilities to other entities, provided the necessary security conditions are met.

With these results, protocols & interfaces for the exchange, negotiation, and execution of mobile service knowledge (effectively constituting mobile intelligence from a system and user point of view), we enable mobile user-centric computing & communication that extends the immediate environment and user experience with optimal and/or meaningful communication, which is not predetermined by any one operator or service creator.

5.1.1 Direction

Following the characterization of MSK in Section 5.1, there is a definite emphasis in Mobile Service Knowledge on the negotiation through communication between these entities and thus awareness of what can be known about other entities through this communication, to satisfy the requirements, which have been listed in sections 1.1.3 and 1.2.

Thus, in comparison to RKRL (see Subsection 3.11.3), XSP adds capabilities for service awareness and negotiation. Similar to RKRL, MSK has principles for capturing cognitive information, but in contrast to RKRL, predefined concepts or a priori knowledge have been made integral components of the language specification only to a very limited extent. The reason for this is simply that MSK is not specifically concerned with radios. MSK is open to the inclusion of a priori knowledge, but knowledge of the world is not an integral part of its concepts. MSK assumes few predefined concepts, primarily a few axiomatic ones (inheritance, causal chains, time, existence; meta-level); much in the same way as object-oriented design has some axioms. While human concepts are not a-priori knowledge in MSK, MSK allows the inclusion of predefined concepts, which correspond to human experience by assigning them to meta-contexts (see below). In summary, MSK is a structural interpretation of how the world in which MSK will be used is conceived, or rather an interpretable structure that will correspond to observable phenomena, simply because of how it is used by the entities that carry it.

5.1.2 Structure and Computability

Regarding the external format as presented by XSP, MSK adheres to some results from W3C, in that a mapping exists between MSK and XML [213], and compatible with the N3 notation [8] for describing logic and relations. The format of Mobile Service Knowledge (MSK) can thus be provided in a short form Resource Description Format (RDF) [214] and internally adheres to the logical notation of the object oriented programming in SICStus Prolog [189]. Objects, i.e., the entities, can reference specifications of remote entities and relations. References to RDF files are allowed. These are followed by type information specifying the external behavior of the agent, as well as inheritance from other entities and relations, and finally tasks, which are goals to be fulfilled after initialization, comprising firing relations matching received events. Thus, entities in our service architecture for adaptive mobile communication will as far as the Semantic Web is concerned be regarded an integral part of this Semantic Web:

“The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.” [9].

On the other hand, MSK (and the service architecture is operates within) has a larger scope than the Semantic Web, as MSK describes more than the interaction between users and systems, but also the causal relations between components within systems.
5.1.3 Example

This section describes an example illustrating important properties of the Mobile Service Knowledge as characterized in sections 5.3 through 5.3.2.1.

As an example, we study the relations governing the negotiations between a user and the access provider in Figure 3-10 on page 42 from the user’s perspective, as shown in Figure 5-1. The relations and objects optionally refer to and inherit behavior from resource description format files. The relation access triggers an event (through the inference in the ACM) because an event is sent to the user via XSP, that an access has been detected by operator WLAN Access Provider 1. The relation access causes the user to query WLAN Access Provider 1 for its profile, if the two have no previously recorded relation(s). The user will interrogate WLAN Access Provider 1 (via XSP) about its relations and thus is able to establish relations to two other entities Credential Provider 1 and Service Provider 1, resulting in the semantic graph of Figure 5-1. In our notation above (adhering to N3 [8]) the access relation is expressed as follows:

```plaintext
/* <http://psi.verkstad.net/msk/relation/access> */
relation @ "<http://psi.verkstad.net/msk/relation>".
relation(access) ::
  type(relation) &
  -> init :-
    that <: xsReq &
    <- xsReq ::
      certifies.init(this)
    <- grant(X) :-
      newIPaddress(X) &
    <- deny(X) :-
      closelink(X)
}.
```

Thus, the access relation expresses the fact that a WLAN Access Provider will state its capabilities, which are to be matched with the user’s requirements and service in question. The inference engine will deduce from the following construction of this graph that in order for the WLAN Access Provider 1 to be able to process its request for access, the user must have valid credentials to certify his/her identity and the WLAN Access Provider must have agreements for Internet access with a network provider signified by the relation deliver, which enables the provider to deliver access to the service - see further the extended case study in Section 9.5.

5.2 Specification

Following this characterization of the properties of MSK, we will now examine the structure of MSK in a formal notation, adhering to XML. It is important to note that these language constructs are intended to reflect the properties, rather than define them. Hence, they most likely will change in future work as language constructs incorporate design choices. Non-language primitives, such as identifiers, are in italics:

```plaintext
Package ::= [RemObjId @ "<URI>"] * (prefix)
  package( PackageId, ExportList[, Options] ),
  [MetaContext | Mobilet] +
```
MetaContext ::= [RemObjId @ "<URI>".]* (prefix)  
  metacontext( MetaContextId [, expiry(DateTime)] ) {  
    type([agent | mis | kb])  
    [ & type(ObjId)]* (inherit)  
    [ & has([MetaContextId | ObjId] (InitValue))]* (collect)  
    [ & has([MetaContextId | ObjId] (InitValue))]* (alt.)  
    [ & state(State)]* (states)  
    [ & ReId RelOp RecObj [ , RecObj]+ [ , weight(W)]]* (ruleset)  
    [ & Task ]* (init)  
  }.

Mobilet ::= [RemObjId @ "<URI>".]* (prefix)  
  mobilet( MobiletId [, expiry(DateTime)] ) :: {  
    type([agent | mis | kb])  
    [ & type(ObjId)]* (inherit)  
    [ & state(State)]* (states)  
    [ & ReId [Guard] RelOp RecObj [ , RecObj]+ [ , weight(W)]]* (ruleset)  
    [ & Task ]* (init)  
  }.

Relation ::= [RemObjId @ "<URI>".]* (prefix)  
  relation( ReId [,Subject] ) :: {  
    type(relation)  
    [ & type(ReId) ] (inherit)  
    [ & Event ]*  
  }.

Guard,
State,
Task ::= 'logical expression' (opt. i/o)

RelOp ::= '->' | '<-' | '</' | '->' | '<>' | '[]' | '-[]' (directed)
| '[]' | '~<>' | '~[]' (temporal)

ReId,
RecObj ::= ObjId.

RemObjId ::= MobiletId | ReId | MetaContextId

Event ::= RecObj << [ ReId[ .task() ] | .Event[ (Subject) ] ] (default)
  RecObj << [ Task | State ]

Task ::= task(ReId[ .event]) (task)

Subject ::= 'logical expression'
  this == sending object
  that == receiving object
  self == the executing object

5.3 General Properties

The structure of Mobile Service Knowledge is equivalent to that of Petri-Nets [13] (directed acyclic graphs) with states and event-driven transitions. This means that within bounded sets of MSK proofs of computability for Petri-Nets apply.

The knowledge representation is based on mobilets corresponding to frames, which are stored and executed or subject to inference in the Active Context. From the above, we can synthesize the following properties regarding the structural entities of the knowledge representation:

5.3.1 Mobilets

External objects must be represented for attaching logic representing the observed events, relations
to other objects, and related behavior. These structural elements representing external objects are called mobilets. Mobilets are placeholders for the aggregation of rules synthesized from observed causal effects (event→action). Receiving such events in the future will thus cause the root mobilet (representing the “self”) to predict causal effects by propagating events in the directed acyclic graph of mobilets and relations.

The semantics of a mobilet is that it corresponds to an observed entity thus in practice it is a software agent speaking XSP, but it can also correspond to a hypothetical object, which in fact does not speak XSP, but is a constant source of data observed by the ACM of the mobile agent and serves as the source of events (e.g., a location service with a reporting function about access points and available data rates) — see further Subsection 4.4.7 on page 60. Optionally, it could be the task of the mobile agent, XSP, and behavior in the ACM to infer these hypothetical objects automatically. Mobilets can have state information corresponding to observed attributes (such as location).

5.3.2 Relations

The semantics of a relation between mobilets is that it corresponds to a capability. E.g., if the relation between the mobilet, which represents the mobile device, and the TV includes the event “off→on”; then the capability represented by this relation includes knowing how to turn on the TV.

Relations are first-class objects in MSK. In the ACM instances of mobilets are synthesized from relation and mobilet templates. This process is described in Section 5.3.3. Thus, relations are an abstraction of rules. Relations are commonly binary, but n-ary relations can be generated as the result of observations of the interactions between several entities — see Figure 5-3.

5.3.2.1 Basic Relations

There are basic (i.e., axiomatic or a priori) relations capturing object-oriented behavior such as “is” or “has” for inheritance or aggregation of the behavior between mobilets. These axiomatic relations are interpreted as the inheritance or aggregation of rules during the assembly of mobilets from the templates (see Subsection 5.3.3 on Synthesizing MSK).

The knowledge structures that the mobile agents contain, mirror the structure of the external communication as experienced by the mobile agents. Therefore, as these agents develop, they internalize the objects that they encounter and attach to this object tuples of events and actions that will be triggered by these events. The objects can and most often will be complex and relations are created to other objects. The relations are of specific types, where the notion of type is formally defined as a relation $r$ over the set of objects $O$, where $x r y \Rightarrow x, y \in O$. The mobile service knowledge primarily defines two basic types of (axiomatic) relations both of which impose a partial ordering of $O$:

- **has** (containment)
- **type** (inheritance, cf. ‘isA’)

Other relations may be defined, such as for instance distance, which is reflexive (i.e., $x$ distance $y = y$ distance $x$). In fact relations can inherit from each other. Relations are most often composite, in that they express both logical relations, leading to a proof process with explicit invocations of communication and inheritance or containment. The arcs in the ontology graph are driven by the events exchanged by the entities via XSP and as a result of the inference in the ACM.

5.3.2.2 New Relations

New (i.e., non-axiomatic or a posteriori) relations between external objects can be synthesized, which are placeholders for aggregated rules representing causal effects. This means that commonly observed relational patterns in rules comprising events-action causal effects between entities can be moved out of the mobilet to these relations. In fact mechanisms in the ACM that continuously monitor
the mobile service knowledge will sieve out all rules that speak of other entities beside the entity itself and move it to a relation.

5.3.2.3 Temporal Logic

Temporal logic: causal chains can include conditions corresponding to a notion of time; thus expressing (propositional) temporal logic [147]. Thus we are able to use the operators \( \Diamond \), \( \Box \), \( \Diamond \), and \( \Box \), to make statements about \( A \) being true always, eventually, so far, or once \( A \) is true, respectively [108]. The importance of this is to enable the ACM using these features of MSK to reason in a space-time domain, and make deductions about states and events in parallel to other reactive and concurrent systems.

5.3.3 Synthesizing MSK

Entities start out with a discovery phase, exchanging their respective (MSK) profiles, after which each one synthesizes an updated semantic graph from aggregated profiles - see Figure 5-2. For instance, two mobilet objects \( O_1 \) and \( O_2 \) sharing the relation \( R_1 \) are instantiated by synthesizing the MSK of their instances out of their templates by copying in MSK from the relation object \( R_1 \).

Figure 9-6 on page 143 shows how MSK is instantiated for mobilet objects \( \text{theo} \) and \( \text{tom} \) sharing the relation \( \text{inet_xs} \) (acquiring Internet Access) from the case study further described in section 10.5 and which further discusses and exemplifies how new MSK is synthesized as mobilet objects discover one another in successive steps. This process is described in the next section.

5.3.4 Context Management

5.3.4.1 Weighting

Weighting of rules involves the synthesis of a proposed set of rules (e.g., using genetic algorithms) from templates, selection or de-selection of the active rule according to the measurements of their respective applicability. Weighting and selection of active rules is optional. The relative success of the execution of rules will be expressed in a change of parameter weights. Rules that acquire a high weight ranking will be selected first and ones with a low ranking will be marked for garbage collection. The ACM temporarily replaces single rules with a rule set, where each rule is weighted and executed in parallel for weighting and weeding out of the weak ones.

5.3.4.2 Expiry

Expiry dates can optionally be attached to relations and mobilets such that those rules or mobilets have not been activated for a certain period of time are marked for deletion (or preferably transferal to a knowledge server via the background handling).
5.3.4.3 History

Another optional attribute for context management is to (optionally) attaching history records to mobilets and relations so that we can track changes (indicated for relation R in Figure 5-3 with an of index ranging over i...j). The history attribute enables a mobilet to improve response to events that do not change in type but change in value, e.g., in Figure 9-5 history records of relations between theo and different access points or the wireless hop, determines decisions by the ACM to assign different priorities to these relations and for instance de-prioritize the GPRS access for other communication than mere signalling.

5.3.5 Meta-Contexts

Meta-contexts can be synthesized from clusters of mobilets. Thus meta-contexts are a structural element in mobile service knowledge. Thus meta-contexts are in essence a cluster of mobilets to which causal rules (or behavior can be attached, in the same way as mobilets. Meta-contexts are in essence a higher-order mobilet. Meta-contexts can inherit or aggregate using the same isA or has relationships as mobilets.

5.3.5.1 Semantics

The semantics of meta-contexts are that they correspond to observable contexts, e.g., ‘everything in a room’, or ‘me watching television’. There is no direct correspondence to a room or a TV, or a cluster of mobilets with a room or TV at its center. The room or TV meta-context corresponds to a synthesized abstract notion of clustered mobilets and relations between such mobilets that we may attach further causal information to it, which we can interpret with our senses and relate to familiar concepts. Thus, the inception of an “I feel good” meta-context would be equally valid as the hypothetical meta-context “X”.

5.3.5.2 Structure

As we have seen, MSK allows the creation of meta-contexts. Meta-contexts are not only allowed to include clusters of mobilets; but also include clusters of meta-contexts. Moreover, meta-contexts are allowed to share these structural entities, thus share mobilets, or even meta-contexts.

5.3.6 Inheritance and Aggregation

In principle, inheritance of knowledge between entities - as per the isA (type) relation - can be an important enabler for knowledge management. Entities can be classified in a taxonomy, such that they inherit knowledge from (multiple) parents. However, taxonomies are defined prior to the creation of types of entities and introduce the need for administration of all entities and their instances. In addition, this also introduces additional complexity regarding synchronization and off-line behavior. On the other hand, copying packages of knowledge, as discussed earlier in this section, creates a problem in that we do not know when knowledge becomes obsolete. Therefore, packages should retain information about origin (e.g., knowledge server identity, location, and profiles) in order to facilitate the wedding out of obsolete knowledge.

In contrast to inheritance (governed by the type declaration) aggregation deals with containment (expressed by the has declaration) as for instance in Figure 5-4 where a person C contains an access (account), a location, but inherits from Person.

5.3.7 Ontologies

Ontologies are sets of concepts and descriptions of these. For instance: in the case of a mobile phone, the concept of a ‘call’ is
central. Ontologies are used in agent systems such as where KIF and KQML are involved in order to ensure the communication discusses the same concepts. Use of such ontologies requires that these are agreed upon prior to use in agents. With XSP, agents carry their own ontology in terms of a semantic graph (cf., an rdf file), showing its direct (potential) relations other agents. This local approach relieves us of having to define a complete global approach, and enables our system to grow dynamically - at the cost of more being communicated (until a meta-level abstraction forms).

5.4 External Properties

Furthermore we can deduce the following external properties regarding inspection and management of knowledge:

5.4.1 Caching

For a description of the foreground/background management see Subsection 6.3.11.

5.4.2 Mobility

Awareness of opportunities for code mobility is meta-knowledge and decisions regarding the transfer of mobilets and/or meta-contexts are inferred and made by the Active Context Memory.

5.4.3 Export/Import

Subsets of the accumulated Mobile Service Knowledge ranging from a single mobilet to meta-contexts at any level can be exported or imported.

5.4.4 Security

In principle, a security attribute can be assigned to each and any structural component. However, from a practical point of view, security and trust are issues between entities (agents). Therefore, security is not given a role in MSK, but taken care of by either the agent or the ACM.

5.4.5 Subscriptions

Subscriptions to mobile service knowledge events are terminated when this knowledge is removed. Naturally this results in a notification to the subscriber. Subscriptions are not part of the mobilets, but registered in the index of all mobilets.

5.4.6 XML

A browsable (via XML) taxonomy of the mobile service knowledge can be generated (as XML) following relations from an index of all mobilets.

5.4.7 Packages

Packages are self-contained chunks of Mobile Service Knowledge for the purpose of transferring Mobile Service Knowledge between entities. An important mechanism for enabling Active Context Memories to apply Mobile Service Knowledge successfully in certain problem domains is by simply adding packages. Packages are thus a construct merely for administrative purposes. Packages maybe assembled off-line through (human) design or assembled automatically by an Active Context Memory (ACM) by encapsulating isolated groups of mobilets (in the sense of having few external relations in comparison to internal relations). Hence, internally in the ACM, packages are descriptions of a set of mobilets. External relations (visibility) are restricted to the mobilets mentioned in the package declaration. Furthermore, meta-contexts consisting of clusters of mobilets are good candidates for packages. Packages can be imported or exported locally or via XSP either manually or automatically. XSP allows other entities to obtain a descriptive list of available packages and import
these packages. See further Subsection 8.3.18 on page 129.

Packages can be *deduced* by an optional mode of operation of the evaluator tool, which maintains a list of cases generated by case-based reasoning and lists the entries as packages. Packages in this case therefore constitute a prototype of service behavior. A default method applied by the evaluator tool identifies central mobilets and the set of mobilets that this mobilet has relations to and creates a package entry listing these mobilets. Examples of these *clusters* are found in the figures illustrating knowledge negotiation and synthesis of Section 9.5.

### 5.4.8 Knowledge Structure

In the following example, a person having certain attributes enters a meeting and is automatically made part of the on-going communication in the room. The room (inheriting relations *enter* and *leave* from mis) would be specified as follows:

```plaintext
/*<http://psi.verkstad.net/msk/space/room>*/
group @ "<./msk/user/group>". (member list)
mcu @ "<./msk/av/mcu>". (conference voice service)
location @ "<./msk/space/location>". (room location)
display @ "<./msk/av/display>". (public display)
mobilet(room) :: {
type(mis) &
   has([display, mcu, location]) &
task(enter.init)
}
```

In this definition of the room, it inherits all the capabilities of a mobile interactive space, but it also has a display, a group of participants (each with their own capabilities), and an actual (physical) location.

In the previous chapter, we also encountered a special case of inheritance for the objects, namely resource description format files, which capture the relational graphs (also called semantic webs) for these objects. Strictly speaking, the agents list is a reference to this relational graph among its properties as described in the previous chapter, but as the objects mirror these agents, then the object can also mirror this pointer to this relational graph.

For instance a person C is an object that inherits its capabilities from a generic ‘Person’ object, which also knows, by virtue of the relational graph that it points to, what relations are possible to other objects. In this case ‘C’ has a point of attachment to the Internet (access with various attributes of its own), as well as a location. Location changes are propagated to the containing object (‘C’ being a ‘Person’) resulting in an action according to this event in this object. Should ‘C’ have other relations, then clearly ‘C’ propagates this event to its containing object, and so on.

```plaintext
/*<http://psi.verkstad.net/msk/user/c>*/
person @ "<./msk/user/person>".
mobilet(c) :: {
type(person) &
   has(location(kista)) &
   access.init(kth))
}
```

```plaintext
/*<http://psi.verkstad.net/msk/user/person>*/
person @ "<./msk/user/person>".
access @ "<./msk/relation/access>". (location)
mobilet(person) :: {
type(agent) &
   has(location) &
   task(access.init)
}
```

The ‘has’ relation between objects in the mobile service is thus reflected as an aggregation of
attributes (in this case location and access) and interpreted by the inference engine to cause them to propagate events.

5.5 Design Choices

Following the choice of format for Mobile Service Knowledge (MSK) - see Section 5.2 - the implementation of the formal language is based on object-oriented Prolog: SICStus [189]. In the case of Prolog, the specifications of mobilets and meta-contexts are implemented as executable objects with clauses.

We can easily extend our system with relations such as has by asserting such clauses when parsing the knowledge structures. Inheritance is derived from such a mechanism that is available in Prolog, while containment is mapped on the aggregation of object references as attributes, in the definition of the object. The formal specification can very well be mapped onto another suitable high-level language with the necessary language constructs. Indeed, a transition from object-oriented Prolog to Oz is perceived as a possible next step due to the more advanced distributed programming language features of Oz. Clearly we also need to extend the knowledge management functionality to monitor that these clauses are removed when the object that they refer to are removed. This task is delegated to the ACM as meta-knowledge. Meanwhile, in SICStus Prolog, objects have a general structure (adhering to the characterization of the properties of mobilets in section 6.2) that can be specified as follows:

```prolog
<object-id>([Argument[<,Argument-1>]*]) :: { 
    [ super(<super-class-id>) & ]* 
    [<sub-object-id>([Argument[<,Argument-1>]*]) &]* 
    //”methods” 
    [<property>]* 
    [<behavior>]* 
    <init-method-id>([Argument[<,Argument-1>]*]) 
}
```

Following the above interpretation in SICStus Prolog of our formal notation, the room, as specified above in Subsection 5.4.8, does not have any behavior except delegating initialization to its contained objects. We can give it behavior for participants who enter, thus modeling the relations to group, display, and persons present:

```prolog
/*<http://psi.verkstad.net/msk/space/room>* /

group @ "<../msk/user/group>". (member list)
mcu @ "<../msk/av/mcu>". (conference voice service)
location @ "<../msk/space/location>". (room location)
display @ "<../msk/av/display>". (public display)
mobielt(room) :: { 
    type(mis) & 
    has([display, mcu, location]) & 
    announce -> person & 
    leave <- person & 
    task(announce.init) 
}.
```

The participants in the group are notified and person(P) is added to the group, while person(P) receives a handle to the group object (sending itself) so that it with the inherited methods can scan the resources in the room and use these. In addition a greeting message is displayed for person(P). As the room delegates this to the display, the display might ask the person what to display relative to this context, upon which the person for instance might send it a handle back to his/her mailbox to display the total of unread messages. Similarly a relevant message may be displayed when person(A) leaves the room.

5.6 Conclusions concerning Mobile Service Knowledge

This chapter has presented a knowledge representation which is machine interpretable and using
inferencing and knowledge synthesis support in combination with XSP (which will be explained in subsequent chapters) allows us to acquire fully operational executable semantic models enabling mobile devices and users to take relevant action in the context that they are used. Importantly this model does not require advance knowledge of the capabilities of other objects and thus circumvents the limitations in UPnP. MSK combines event-driven mobile-code with semantic graphs that are accumulated during the life-span of an ACM. This constitutes an important contribution beyond what previously has been possible with KQML/KIF [46,180] or its XML-based successors DAML+OIL [166].

What has been achieved here in comparison to these previous results is that MSK is able to express dynamic behavior in fully operational models, combined with its capability to express semantics between the constituent entities. MSK allows the Active Context Memory to internalize the (external) entities in the communication space along with the relations between them and at the same time constituting executable code. KIF is a very expressive language comprising a collection of machine interpretable logic statements about the world, which does not specifically address encapsulation of external objects in a similar sense as MSK does. Moreover, KIF has no real concept of event propagation, which is specifically addressed by MSK, allowing the ACM to simulate the external world. DAML+OIL extends the tradition of KIF to XML [213] and RDF [214], but has stronger support for expressing object and relations, the purpose of which is allow programs to deduce facts and reason about these objects and relations. Neither KIF or DAML+OIL is particularly concerned with the dynamic behavior between objects, as MSK is; as a consequence, neither are particularly well suited for our purposes.

In summary, MSK enables us to express executable event-driven dynamic behavior between objects, thus allowing the ACM to internalize a fully operational model of the perceived external entities, also enabling it to simulate scenarios between them, and thus deduce reasonable strategies for its communication, relative to its purpose as specified in MSK. With this, MSK enables us to design the eXtensible Service Protocol (details in Chapter 7, which can now be agnostic about the services which will be negotiated between entities.
This chapter defines and examines the external operational properties of a personal active context memory. In light of the ‘communication space’ paradigm, as proposed in this dissertation, we examine the means by which a personal communication agent can provide us with true ‘intelligent’ communication.

6.1 Overview

The overall purpose of an Active Context Memory co-located with a agent (hereafter referred to as ACM) is to capture the user’s and/or device’s behavior in order to:

- Adapt communication to a change in type of resources.
- Adapt communication to a change in available resources (i.e., becoming available or unavailable).

The Active Context aggregates and stores Mobile Service Knowledge (MSK). Mobile Service Knowledge may be received from other entities. This knowledge can be used to register information about events & responses from other entities or events & responses from the agent itself. As we have seen in chapter 6, the MSK is organized as rules in a semantic graph of objects mirroring the external objects that the agent has encountered. The evaluation of MSK involves inference on the preconditions of the rules belonging to the object that generated the event. When the rules are triggered (i.e., the preconditions are met) this can causes not only a change of the state which the agent is in, but also cause events, or perhaps bits of Mobile Service Knowledge to be sent to other agents - that is, carried by XSP.

This chapter examines events occurring on two levels. One is between agents, and the second is between objects inside the agents (Figure 6-1). Events sent to mobilets that correspond to agents are forwarded to these agents over XSP. Therefore, there is a correspondence between the communication environment as perceived by the Active Context Memory and the state of the semantic graph of Mobile Service Knowledge. Evaluation of the contents of this graph and inference on the logic of its objects and relations determines responses of the ACM to communication events. As the meta-contexts of Mobile Service Knowledge refer to mobilets corresponding to an ACM belonging to another entity, the ACM may decide to relocate this knowledge to those entities. Thus, we create a vehicle with knowledge capable of reasoning about itself.

Figure 6-1. ACM, XSP Events
6.2 Machine Learning

Knowledge that is stored in the object may either be inserted by a programmer (e.g., in domain specific packages - see Subsection 5.4.7 on page 86) or be the result of learning processes. The learning mechanism works as follows: Generate a number of prototype semantic graphs extending the existing semantic graph with hypothetical entities. The ACM keeps a log of past events. Out of this aggregated collection of events the ACM looks for patterns (repeated occurrences of events from certain objects). For instance, if a user A is contacted by other users B and C in a certain location, then a series of hypothetical mobilets M1, M2 are created grouping and aggregating B and C. If this location is revisited then A's repeated responses in this location are remembered and later recalled.

The design and detailed properties of these learning processes are outside the scope of this dissertation, but it is important to point out that the Active Context Memory should incorporate such mechanisms, which automatically synthesize contextual information from the events that it receives. [111] describes how cognitive processes in a mobile device can accumulate patterns and synthesize ontologies (i.e., a collection of relevant concepts regarding the mobile communication, and thus may be used to automatically generate service knowledge in a Radio Knowledge Representation Language. This approach can be applied in a similar fashion to mobile applications. Thus future work should focus on examining the detailed properties and design requirements of functionality for enabling mobile applications to learn new concepts regarding a certain problem domain, be able to relate events to these concepts, and thus become aware of relevant changes in the communication environment. A Machine Learning mechanism for the Active Context Memory should synthesize new Mobile Service Knowledge (MSK) from chunks of MSK or events that it receives via XSP.

The limitations governing systems reasoning about themselves have been defined in work by the logician Gödel showing that a logical system of arbitrary strength cannot contain proof of its own completeness or consistency, corresponding to Turing’s halting problem. The theoretical background following the proof by Kurt Gödel [49,50] is that a logical system of arbitrary strength (as formed by the ACM and MSK) can express sentences such that neither this sentence nor its negation is a theorem of MSK. In this context, the practical interpretation of these limitations is that the ACM must be able to learn new concepts and resolve them as extensions to the inference (alterations to the algorithm of the inference engine) and extensions to the Mobile Service Knowledge grammar. Hence, externally generated knowledge (by events that are new to it) will allow the ACM to extend its reasoning processes, and jump out of its closed system.

6.2.1 Example

In this section, we will outline the properties of such machine learning mechanisms, by extending an example taken from Chapter 9 illustrating these necessary properties: The Mobile-Aware Media Player is aware of the existence of Mobile Proxies in WLAN access networks and GPRS access networks, whose purpose is to do predictive caching based on user profiles in order to optimize the download of multimedia over the wireless links, and to minimize the consumption of resources. One day, the Mobile Aware Media-Player may learn about new types of access networks. One such access network is the Mobile Backpackers Free Radio Network (MBFRN), which has a distinctly different approach regarding the delivery and routing of content in that it uses personal pico-cells relaying information between peers. These peers are in the possession of mobile devices with considerable storage capacity. Therefore, every member’s end-device becomes both a client and a mobile proxy. The wide-area GPRS interface can be used to relay information about the location of the user, direction, communication conditions, and content of the mobile device.

6.2.2 Requirements

From this example we can deduct some general requirements regarding the capabilities of machine learning in the Active Context Memory. First, it must be able to learn by example or do case-based reasoning. The notion of meta-contexts in Mobile Service Knowledge is an important enabler for
learning by example as the Active Context Memory is able to import a meta-context comprising such a mobile proxy for the MBFRN and add it to its knowledge. Thus the entity must be shown this example by another entity that has it. From this we deduce a requirement on the ACM to look for capabilities in the ACM of other entities, such that it is able to solve a certain task. Therefore, it needs to be able to do reasoning about the capabilities that are being offered to it, to determine if they can assist in solving a certain task. Secondly, when the ACM has accomplished this task (in the example, how to use the MBFRN in order to download content and upload content to those who need it), it must be able to learn how to optimize these tasks using genetic algorithms relative to these tasks in order to assess the success of strategies.

6.2.3 Components

The ACM can optionally include machine-learning mechanisms, such as a concepts generator (genetic algorithms) based on received events and Mobile Service Knowledge. The generated concepts are measured, then selected or discarded according to their relative success, due to the addition of an evaluator tool, comprised of genetic algorithms, to the inference engine, allowing it to measure the success of the knowledge, and thereby allowing the knowledge to add to and alter the behavior of the genetic algorithms. From an engineering point of view, we must ensure that the enhanced inference of knowledge is able to determine an answer within a bounded time. This bounded time is achieved through a time-out as a catch-all mechanism, which triggers in case that the inference is not able to arrive at a decisive answer. The ACM decide to time-out the inference depending on the urgency of the operation but it could also ask the user for advice. If the service involved concerns the intermittent download of content, then other factors will be more important for determining the urgency, for instance battery level. A knowledge integrity checker, analyzing new additions of knowledge to the semantic graphs ensures that no cycles or circular references will be introduced.

**Long term vision:** Given the learning mechanisms that are mentioned in this section we can envisage the creation of mobile robots and mobile brains, see further Subsection 9.3.2.

6.3 Architecture and Design

The agent has to interact on one hand with other agents and on the other hand with the services and resources that need to be involved in the communication. The agent handles the communication management (naming, localization, session invocation, mobility support, etc.), but the agent also inherits XSP classes for handshaking with services and resources, which enable the agent to subscribe with other agents, exchange service knowledge, and propagate events. The eXtensible Service Protocol (XSP) is explained in chapter 8. The agent delegates the tasks concerning the aggregation and management of Mobile Service Knowledge to the Active Context Memory (ACM). The internal structure of the ACM consists of a knowledge base, an inference engine, a garbage collector, and foreground/background handling - see Figure 6-2 and Figure 6-3.

The external properties of the components shown in Figure 6-2 and my design decisions are explained in the next few sections. Their internal properties are explained in chapter 6, Mobile Service Knowledge (MSK).

6.3.1 Knowledge Base

As we have seen in Chapter 5, the Knowledge Base is structured as a semantic graph of objects (see
Figure 6-3. Personal Mobile Agent design

Figure 6-2, Figure 5-4 on page 85, etc.). The objects mirror the objects that the ACM has observed. A mobilet is a container for clauses (rules) pertaining to an object or a cluster of objects. For instance, the ACM also contains an index of mobilets, allowing it do fast lookups of the objects. In Figure 6-1 the agent A receives an event e(B) from agent B by means of XSP, which is delegated to the ACM of agent A where it is forwarded to the mobilet representing agent B resulting in a new event e(B→A) - a symbolic notation for an event caused by B and received by A.

6.3.2 Knowledge Management

Knowledge Management is needed for user agents to retain contexts. Two problems arise as knowledge grows over time. Searching large knowledge bases takes time and such a large knowledge base has a large memory footprint, which is undesirable in thin clients. Therefore we must be able to scope queries pertaining to a specific context (looking for particular meta-contexts as per Subsection 5.3.5 on page 85), and we may also move knowledge that is seldom used into the background (as per Subsection 6.3.11). In addition, knowledge that has been invalidated or is no longer referenced must be removed (Subsection 6.3.6).

Knowledge Management thus involves optional components (such as an inference engine, machine learning capabilities (Subsection 6.2.3), garbage collection, and foreground/background knowledge handling) and mandatory component agent communication.

6.3.3 Knowledge Extension

Extension of the Mobile Service Knowledge between agents is easily achieved in this model, as agents will detect dissimilarities in the service repertoire between them and thus can exchange mobilets using the XSP protocol. However, such exchange presupposes, as we have seen earlier, a recreation of the semantic network of mobilets in the receiving agent. Thus, potentially, a received mobilet might not have the exact same semantics in the receiving agent as in the agent where it resided earlier. See further Subsection 5.3.3 on Synthesizing MSK and Subsection 6.3.10 on Reconciliation of knowledge structures.

6.3.4 Knowledge Scoping

The knowledge representation needs to have 'context keys' which allows agents to asked scoped queries pertaining to a particular context (Figure 6-4). In turn this means that a placeholder for context is needed that encapsulates (pointers to) knowledge. Clearly, the notion of packages and
meta-contexts offer organizational principles for the ACM to look for specific contexts. As new entities appear, new placeholders for knowledge (mobilets) are created in the Active Context Memory. The Active Context Memory will store events as relations between mobilets, and thus when events occur it is able to recreate the behavior corresponding to previously occurred events and events occurring in meta-contexts that involve multiple entities.

When certain facts apply to different contexts (as meta-contexts may either share meta-context or mobilets below them), these facts must be duplicated. A second approach is to use pointers to knowledge. This optimization requires more management to keep track of what context keys refer to specific knowledge. In addition, more computation is needed to decide what knowledge can be moved to knowledge servers.

Even more importantly, the Active Context Memory needs guidance to decide what context keys to choose. Preferably, these choices should be made without requiring input from the user, as our objective is to create transparent adaptive services. Therefore a good choice is to initialize keys for any entity that the user agent encounters and informs the Active Context Memory about (e.g., a printer, a public display, etc.). At this granularity the designs scales well, as computations for garbage collection and foreground/background-handling of knowledge do not become voluminous with maintaining counters for referencing of knowledge, this parallels caching and memory management strategies in symbolic languages, such as Prolog.

6.3.4.1 Scope for Managing Obsolete Knowledge

Scopes are dynamic concepts. In this example scope is used for managing obsolete knowledge. In Figure 6-4, if there is a display and an event occurs in the room, that involves the display, this event is recorded in a clause (asserted) with arguments room and display in its head, besides sending the events to the object so that a temporary local scope is available.

The structure of the dynamic knowledge in Figure 6-4 for scoping as discussed above needs special attention. The logic clauses pertaining to the display when this device is turned off, will be unused and thus (D), (R,D), (P,D), (R,P,D) will no longer be referenced. A garbage collector needs to traverse the knowledge base and check whether these clauses are still referred to by an active entity. As this is not the case for the (D), (R,D), (P,D), (R,P,D) groups, references to these clauses will be passed to the foreground/background knowledge handler to move them from ‘primary’ memory to a knowledge server (as described in Section 4.7). Such a foreground/background knowledge handler mechanism should not be too aggressive, as otherwise it would migrate knowledge to soon, as the user may turn on the display soon again. In that case, the foreground/background knowledge handler should delete the references to (D), (R,D), (P,D), (R,P,D) from its list of candidates for migration - for a discussion of foreground/background knowledge handling see Subsection 6.3.11.

6.3.4.2 Multiple Stacks for Managing Parallel Contexts and History

Furthermore, we may need to manage parallel contexts and their histories respectively. This is important because the entity (for instance a user) might be doing different unrelated things, and need to trace and retrace the history of the current context in order to recover previous states of negotiations. This way, a user can negotiate the delivery of a service while simultaneously but
unsynchronized negotiating an entirely different service, e.g., the exchange of a multimedia object. The history mechanism is also useful for minimizing response times in and between contexts.

Figure 6-5 shows an example in which a user (self) discovers a MIS in which the user chooses two services from the ones that are available. Consequently two parallel context frames are pushed to the stack where in the left branch of the stack the user negotiates with a proxy that points to an application provider (ap) which in the next frame that is pushed on top of the stack establishes relations with the user in order to deliver the service via the proxy. In the right branch the user discovers another user who points at a multimedia object in the MIS. During the discovery and negotiation of new services these stacks grow with each successive step, not all of which need to be saved. The Garbage Collector can optionally remove intermediate frames, when there is a shortage of memory. The penalty is losing the history of the negotiations — for a more detailed example, see further Subsection 9.5.5 on page 150.

6.3.5 Concepts Generator

In this section, we examine several models for synthesis of new concepts, their maturity for use in mobile networks, along with their potential and limitations for creating useful or even cognitive services, and comment on the current and future direction for concept generation for Mobile Service Knowledge in the context of this service architecture:

6.3.5.1 Synthesis from A-Priori Knowledge

This approach is similar to the one in [111] for generating new concepts, applying a pre-specified taxonomy for mobile service knowledge in a specific domain, for instance opportunistic use of heterogeneous access networks for the transmission and reception of smart multimedia packets. Model-based reasoning is applied to generate and evaluate alternative concepts. Domain specific meta-contexts may evolve over time, even suggested to the user. Likewise user input may be tapped and processed and generate other concepts that can be linked to the a-priori ones, and thus be the basis for the creation of new meta-context. The premise that a-priori concept will be used defines its power but also limitation, as (unless some other mechanisms are added) the rate of concept growth will be moderate.

6.3.5.2 Outline of the Concepts Generator

The general outline of the concepts generator in the ACM is the following is an interpretation of the approach described in paragraph 6.3.5.1, in that it includes only a minimum of (predefined) a-priori: concepts and that domain specific knowledge can be imported as MSK. Hence, the approach is more open and a step towards the approach in paragraph 6.3.5.1, albeit a pragmatic one based on the observation of the limitations of systems that can be built today.

The task of the Concept Generator is to generate new meta-contexts that have explanatory power relevant to observed events or Mobile Service Knowledge (MSK). The Concept Generator monitors new events or MSK changes. Known events or MSK changes cause no immediate action. However, in the background the Concepts Generator continuously performs conceptualization tasks, analyzing the relations between mobilets or meta-contexts in the semantic graphs of the knowledge base.
more than two mobilets or meta-contexts develop relations (by sending events or Mobile Service Knowledge changes), than several provisional meta-contexts can be synthesized with these objects as elements, with different relations. Should the meta-context no longer be used then an expiry time will tell the garbage collector to remove the meta-context entirely. A selection is made between proposed meta-contexts upon receiving additional events or new meta-knowledge, where suitability (i.e., the best match will be rewarded when selected). The generated meta-contexts are provisional conceptualizations of observed relations between clusters of entities. Therefore, there is a decay property attached to this knowledge, which is reduced at regular intervals leading to a half-life, where knowledge below a certain threshold is collected and removed by the garbage collector. The decay property is reduced if the knowledge is actively being used. The half-life is thus related to the relative usefulness of this knowledge. As the expiry date is a simple timestamp it only measures whether the knowledge had been accessed at all within a given time frame.

This model is able to generate meta-contexts, where their relative usefulness is measured by the methods described further in 6.3.8.

6.3.5.3 Example

Figure 6-6 shows the relations between a user who registers with an applications provider, who refers to a local access provider, who in turn also has a proxy which can deliver or complement content. Then in order to make intelligent decisions about resource contracts between the user and the service provider’s proxy the concepts generators in the ACMs of the user and proxy an infer and synthesize extensions to the use relation in either ACM. These extensions are inferenced from resource attributes and preferences found in the existing MSK of these entities. Should a preference be tagged for user input, then the ACM may query the user. The existence of these extensions are flagged for reconciliation (see Section 6.3.10), and result in a negotiation facilitated by XSP, during which the ACM inferences the resolution of unknown references from user input and/or values from the other ACM, during which potentially new relations may be established. The resulting resource contract will thus for example regulate the minimum data rate needed for specific types of content, when they should be delivered, the cost for the transmission, etc.

Evidently, this is an area where packages with domain specific Mobile Service Knowledge can further simplify the negotiations by simply being imported and reconciled.

6.3.6 Garbage Collector

The extension of knowledge (see Subsection 6.3.3) calls for a knowledge garbage collector, which removes unused contexts and knowledge (Subsection 5.3.4 on page 84). Garbage Collection is a continuously running internal process. It monitors which mobilets or meta-contexts (and thus clauses) are no longer in use. Depending on the configuration of the ACM this data is simply deleted or marked to be moved to a Knowledge Server by the Foreground / Background Manager.

Garbage collection can identify knowledge that has passed the expiry date or knowledge (rules) that does not have enough weight to be kept in memory relative to the priority required (due to memory, power constraints, etc.). This means that mobilets can become isolated having no impact on other mobilets or vice versa; this knowledge can be detected and collected. Garbage Collection of Meta-Contexts is conducted similar to that of mobilets. Packages (see further below) are allowed to remain in the ACM even if they contain mobilets that are not referred to by other (external) mobilets, until they expire (see Subsection 5.3.4).
6.3.7 Integrity Checker

Structural Integrity, Semantic Integrity, and Integrity of Purpose are maintained through applying formal and pragmatic methods. Mobilets without relations and relations (rule sets) pointing to non-existent mobilets are removed. The pragmatic method is to start the search beginning with the mobilet marked as the self (Subsection 5.3.4) and recursively work outward and removing ‘dead’ branches. See further Chapter 6 and Subsection 8.3.2 on page 122 on Feature Interaction.

The Integrity Checker is a service in the Active Context Memory, which may either run in the background or be called when changes in Mobile Service Knowledge occur. Its purpose is to facilitate the removal or addition of Mobile Service Knowledge, without jeopardizing its evaluation (due to loops or unresolved references). For a given set of MSK, it will traverse the Directed Acyclic Graphs (DAGs). First it will find any unresolved references and list trajectories and thus be able to detect loops. Second, it will try to find a close match and repair unresolved references. If loops are found (a mobilet or meta-context, which either has a ‘has’ or an ‘isa’ relation with itself), then the last branch which closes the loop is removed from the list of traversed mobilets which the Integrity Checker maintains (i.e., the list from the latest central mobilet that the Integrity Checker was recursively currently checking). Third, unresolved references are removed.

6.3.8 Evaluator Tool

The purpose of the Evaluator Tool is to measure the success of the Mobile Service Knowledge and select Mobile Service Knowledge (i.e., mobilets or meta-contexts) that has displayed a greater degree of correspondence to experience. Measurement of success can be observed using different methods.

First, the Active Context Memory may receive notifications from the receiving entity of the successful activation of an entity corresponding to the mobilet or meta-context. This is rewarded by an increase of the credibility attribute (C) to S[uccess] %, as credibility ranges between 0% and 100%. Failure will result in a decrease by F[ailure] % — the variables (S and F) and choice of their values are discussed below.

Second, the Evaluator Tool keeps track of the relations of mobilets or meta-context (i.e., the subset of Mobile Service Knowledge being involved in the response to an external event. Since these strands of MSK have cooperated in a successful execution, this causes the Evaluator Tool to step up their credibility with R[epeat] % increases. Failure causes the credibility attribute of this subset of MSK to be decreased by R %.

Finally, as multiple meta-contexts (generated by the Concepts Generator) will be involved in the response to an external event (and thus share mobilets), the credibility attribute of these meta-contexts that have been deselected will receive a R % decrease in credibility. Zero or negative credibility causes the item from the MSK to be handed over to the Garbage Collector.

The Evaluator Tool will track the successful execution in terms of correspondence with the real world (possibly resolved through involving the user or otherwise). This tool measures the explanatory success through weighting and selecting by means of genetic algorithms. Hence the Evaluator may change the values of S, F, and R over time, based on these measures.

The choice of values for variables S, F, and R are based on the observation that their combination reflects the success, failure, and repeated success or failure (hence reinforcement by increments) which takes place during evaluation of the Mobile Service Knowledge. Thus, a reasonable strategy for our design is that successful activation is awarded high credibility (C ← S), failure results in low credibility (C ← F), and repeated success or failure in increments or decrements of the credibility (C ← C ± F).

The choice of the “good” initial values of S, F, and R, and how their values should change, based on measured explanatory success using genetic algorithms is a subject for further investigation.
6.3.9 Inference Engine

The Inference Engine (see Figure 6-2) is considered to be meta-knowledge relative to the Mobile Service Knowledge. The Inference Engine is a service, which is invoked by external queries or internal events. The Inference Engine evaluates the clauses (rules) of the mobilets in the knowledge base. Hence, it is the mechanism, which causes events to be propagated between mobilets within the Knowledge Base. The Inference Engine is an optional intermediary between knowledge queries and the actual knowledge, which can be used to allow more powerful queries. For instance, instead of only being able to ask for specific facts, the inference engine can derive answers by combining multiple facts and rules (with variables). When an external event is injected into the Mobile Service Knowledge, the Inference Engine looks for a matching relation from the mobilet corresponding to the sender (entity). If the event or the entity is novel, this information triggers the Concepts Generator to produce a relational graph that is added as new Mobile Service Knowledge.

In its traversal of MSK, the Inference Engine will encounter parallel structures that match and produce answers, as a result of the Concepts Generator producing multiple interpretations, which will each be evaluated for survival by the Evaluator Tool. The final selection of what event to send to one or multiple entities is based on whether the inference converges in one or several mobilets.

6.3.10 Reconciliation of knowledge structures

Here we will study in more detail how the exchange and negotiation of mobile service knowledge results in building up and reconciling knowledge structures in the mobile agents.

As an object detects the presence of a cluster of objects that are arranged to communicate via XSP, then the outermost containing object is selected according to the has relation, upon which we need to establish a new ordering of the objects according to the relations at hand. This is illustrated in Figure 6-7.

6.3.10.1 Step 1

First the agent A announces itself to ‘B’ (assumed to be the top level container of the cluster objects). Then B has to follow different paths depending upon whether A is a new object or a known one, and depending on whether an rdf file has been created for it and is available on the Internet, a provisional local rdf file (containing the semantic graph of objects) must be located or as a last resort created. The reason for this backup strategy is that we cannot always be assured that Internet access

![Figure 6-7. Reconciling Knowledge Structures](image-url)
is available even in the case of local connectivity between mobile artifacts. The local rdf may contain no relations at all, in which case no harm is done other than that the object does not relate to anything in the on-going communication (in which case it does not participate at all). This can be the result if B fails to recognize familiar properties of how A should be used in the profile that is sent by A. Such recognition could be accomplished by a set of common sense rules or a genetic algorithm that can be improved during the life span of this agent.

6.3.10.2 Step 2

Once the rdf file is available a relational graph of the aggregated clusters of objects must be created or recreated in the case of a reappearing object or cluster of objects. Once this has been accomplished the internalized objects (i.e., the mobilets) are initialized. Such initialization functions as a tentative instantiation of the agents that they represent, thus instantiation of the person ‘C’ (see Figure 5-4 on page 85) may result in its agent actually looking for a location agent and an access network, and may or may not find it.

6.3.10.3 Step 3

Therefore, in Figure 6-7, the initialization of mobilets is followed by another {announce} message. Thus a cluster of objects will self-configure as an application-level active network.

6.3.10.4 Example

When Person C (from Figure 5-4 on page 85) approaches a meeting (see Figure 6-8), then these structures need to be merged and reconciled. In the general case, the outermost containing objects in both structures with respect to the has relation, will try to attach to each other. Then contained objects are tried in the next step. The semantic graph in the RDF file for C (which Person refers to) contains a reference to a latent Group object. The result of which is that C attaches to the group and both initialization, resulting in C being invited to the on-going communication. The result of this merging process is depicted in Figure 6-9.

In this section we have seen that when an agent appears (a service or a partial service) one or several new entries are created or activated for the objects that it encounters. For instance, when a mobile interactive space (e.g. a virtual meeting) agent is detected with a number of participants, then the active context agent registers and also obtains the code for invoking communication with other participants from the mobile interactive space, by delegating the announcement and propagating other events to the contained objects. In the next section, we will study the knowledge representation and its properties in closer detail.
6.3.11 Foreground / Background Knowledge & Handling

Foreground/Background Management is a service, which continuously monitors which knowledge should be moved to or from a Knowledge Server. It runs as a background service. However, it can also be forced to run, for example, when the mobile device runs low on memory. As the knowledge content of the Active Context Memory grows over time, a large portion of this knowledge is less and less frequently used. In such cases, a strategy is needed to reduce potentially increasing knowledge search times, as well as bound the footprint of this knowledge (least it require more memory on the mobile device).

Subsection 5.3.5 on page 85 shows how knowledge can be partitioned related to contexts, with information on the frequency of how often it is used and when it was last referenced. Based on this information, the garbage collector can inform the background-foreground handler about partitions of the knowledge base that can be moved to a knowledge server (Section 6.5), and retrieved when they are needed again, as references to removed knowledge are kept, and when exercised, retained again. Figure 9-7 on page 143 shows examples of knowledge (in shades of lighter grey), which has been deleted but indirectly available. This server may in turn decide when it is time to erase this knowledge by attaching an expiration date to this knowledge, unless the knowledge is requested again.

The physical distribution of knowledge is such that context placeholders and the knowledge that they hold or refer to can be uploaded to knowledge servers, using XSP as explained in the previous section. The question arises as to whether the user should be aware of whether he/she wants to upload what he/she knows or if this should be done transparently. I propose relieving the user from these migration/paging decisions, thus the knowledge garbage collector simply requests that less frequently used knowledge be moved in the background to knowledge servers, therefore minimizing search times and client footprints. Much of the time such background transfers can be delayed until there is lots of communication capacity.

6.4 Design Decisions

The agent is merely a point of integration of the GUI, VoIP, SIP, XSP, and the ACM. As a design decision Java was a good choice for implementation language, because it supports object-oriented programming styles, supports communication between objects on a network, and facilitates the inclusion of a simple GUI, all in one software package. The behavior that is present or envisaged in the agent provides service integration. However, Java is not indented to directly express or evaluate logic statements. Therefore Mobile Service Knowledge was expressed in Prolog and the agent could interact with the service knowledge in the ACM via the Jasper interface [179]:

As a design decision, following the use of Java to implement the agent, a Java API for SICStus Prolog [189] was used to interface the Knowledge Base, which was written in an object-oriented subset of SICStus Prolog. The interface between the two is available under the name Jasper in SICStus Prolog [179]; hence the use of the concept of ‘clauses’ to describe individual pieces of logic. Jasper allows us to run under Java. A possible future direction of this work is to port this to Mozart/Oz.

KQML was used for agent communication in the early prototypes (see Section 9.6). In later prototypes KQML has been replaced by XSP, this is described in further detailed in the Chapter 7.

In SICStus Prolog [189], an object oriented encapsulation facility for knowledge (logic clauses) is available. Prolog meta-contexts can be mapped onto parameterized clauses (thus with at least N parameters for N mobilets), where the evaluation of the body results in forwarding the events to the objects involved as outlined earlier in this chapter. These clauses can be asserted or retracted depending on the occurrence of communication events that involve these objects. Therefore we are able to use these clauses as a means to limit the scope of behavior to a limited number of objects in a space without involving all of them.
6.5 Mobile Knowledge Repository

Figure 4-8 shows the operational environment for a Mobile Service Knowledge repository (KB Server). XSP enables entities to locate entities that are Repositories for Mobile Service Knowledge (MSK) and have a profile with a \(<db>\) attribute. MSK Repositories offer a foreground/background MSK management service (see Chapter 6.3.11) enabling other entities to off-load or even host the content of their ACMS, i.e., you should be able to have your active context memory hosted in the same way as your home page, by an application provider. There is no requirement that this should be bundled with the network access. Each of these contexts can be segments of a larger “brain” (Section 9.3) or knowledge base. This approach opens a new perspective in that given a sufficient level of trust, my computing and communication actions might be guided by a body of knowledge that is shared by a larger number of users. Although the system can learn more quickly, it can also be subject to false knowledge attacks; this is a topic of future research (Chapter 11).

6.6 Conclusions concerning an Active Context Memory

This chapter presented an Active Context Memory, which provides knowledge management, and interpretation support, which in combination with the Mobile Service Knowledge presented previously, enables entities to exhibit awareness of other objects in their communication environment and take action according to their perceived context. Hence, the ACM leverages the features of MSK by interpreting, extending, and managing MSK in order to create the dynamic behavior that is needed. ACM in combination with the eXtensible Service Protocol enables entities (resources, users) to negotiate and act in concert and achieve a common task within the context of a user or an application. Equally importantly, the ACM encapsulates knowledge and reasoning (which together constitute mobile intelligence) enabling the uncomplicated extension of and integration with existing SIP User Agents and subsequent deployment in large-scale (mobile) networks, in order for end-devices and resources in the network to exhibit the adaptive behavior with respect to the user or communication context, which is the objective of this dissertation.

From a user perspective, the extension of entities with an ACM, which is able to acquire, synthesize, and reason about a service behavior modelled in MSK, means that service behavior is created without requiring users to actively program or design their services, as previously has been required from users in CTI and IN systems. This relaxation of requirements on the user and creation of automatic invisible service creation is one of the important contributions of this dissertation. Finally, as the Active Context Memory records & manages user contexts and monitors user input it is able to decide from existing MSK whether a user action probably will be successful and can suggest an alternative (optimal) response for the user, as all this user information already is available. Even more importantly, the ACM in combination with MSK anticipation of reasonable communication behavior of entities will reduce or prevent unnecessary claiming of communication resources, enabling us to further optimize our solutions.
7 An eXtensible Service Protocol

In this chapter, a novel eXtensible Service Protocol (XSP) is described. Its properties and contributions are examined in relation to two well-known protocol suites for pervasive computing JINI and UPnP. After examining its properties, we look at how it can enable ad hoc mobile applications, as well as enabling applications to adapt to the conditions and context of the communication.

7.1 Purpose

In the following sections I introduce the eXtensible Service Protocol (XSP) as an open vehicle for negotiating communication between parties (see Figure 7-1). XSP is designed to avoid the shortcomings and the limitations in the systems that were discussed in the introductory chapters. XSP does not rely on network services that are packaged with a particular type of access, and allows us to move the point of service integration to the end-points. Given the support for addressing, naming, and localization of resources that is offered by SIP, the eXtensible Service Protocol facilitates the discovery of services, event handling, and access to the services of other entities. Similar to UPnP no byte code is necessarily sent between end-points. However, the devices do not advertise themselves to control points. As we shall see in this chapter, XSP offers a direct mapping between entities and services (unlike UPnP which has no direct mapping between control points and services), without relying on the registration of these services in central servers (as is the case with JINI). As such, XSP facilitates the use of services between end-points in a scalable way, which is not possible and manageable through the use of either JINI or UPnP. Hence, XSP enables mobile networks to evolve into large-scale open systems of cooperating devices and resources, such as public mobile networks.
Even more importantly, in addition to the services similar to those present in JINI or UPnP, which were mentioned above, the eXtensible Service Protocol (XSP), enables entities to negotiate services. Given a certain request, the entity will respond by sending a relevant subset (initially pertaining to the immediate context of the mutual relation between the mobilets present in the Mobile Service Knowledge (MSK), which is present in the ACM (Figure 9-9 on page 144 provides an example of this). This subset is then matched to the immediate context of the mobilet corresponding to the sending entity in the semantic graph of the receiving entity. Hence, the receiving and sending entities will be able to agree on a common model for the task in question. Hence, XSP is a set of capabilities merged into one protocol. These capabilities are service discovery, event handling, object access, and negotiation.

7.2 Steps in Negotiation

This section further details how the agents use the XSP protocol. Figure 7-2 shows an overview of the protocol stack and the location of XSP in it. The mobile agent handles SIP and XSP. The agent handles the (de-) registration and keep-alive messages, but then hands the XSP based communication over to the Active Context Memory (ACM), following these general steps:

Discovery In order to find XSP enabled entities, multicast requests are sent on a well-known XSP-port. Knowledge of the existence of entities may already be in the Active Context Memory. In this case the agent merely re-registers. It registers with implicit subscription to events (either generated by triggering relations between the corresponding mobilet in the ACM of the entity with which the agent has registered or requests for the exchange of Mobile Service Knowledge). Query profile: type, event/actions, states, (and neighbors). If the agent that has been found is a communication space, then it tries to register with the other agents that this communication space already knows (repeat steps 2-7 below).

Event Routing If the entity that is found is a sensor, then the agent that registered with this sensor, just receives (i.e., it does not send except for keep-alive notification) events, otherwise, if the entity found is a repository (and not a mobile interactive space, the two not being mutually exclusive), then it invokes foreground/background knowledge handling for this object. It also sends keep alive messages to other agents with which this agent is registered.

Extension of the agent’s MSK does not result in sending an event to its registered agents. However, these agents may or may not request to be notified of new capabilities, provided they have sufficient rights for claiming this information.

Negotiation Analyze the Resource Description Framework (RDF) file and select a matching object profile. The ACM looks for familiar semantic patterns. If there is a match, then the querying agents each request the event/actions that it does not have in its repertoire.

If the ACM is unable to instantiate the received profile in its own knowledge base, the profile will be presented to the user (if there is one) for input, or else will store this profile for later instantiation due to new user input or events. The ACM can also respond to the presence of agents (e.g., detection of a public display and/or (stereo) speaker(s) it will trigger a video conferencing dialog).

Negotiation of the communication is carried out according to the following algorithm (in pseudocode):

1. Find XSP enabled entities:
   Multicast requests on a well-known XSP-port.
   Local sensors on the mobile device are discovered by simply looking for their declarations, e.g. via a configuration file
If knowledge of the existence of entities is stored and present in the Active Context Memory then the agent merely re-registers.

2. Register with implicit subscription to events.

3. Query profile: type, event/actions, states, (and neighbors)

4. If the entity that is found is a sensor then stop here; just receive events,
   else if it is a repository then invoke foreground/background knowledge handling with this object.

5. Analyze the RDF and select a matching object profile. The ACM looks for familiar semantic patterns.
   If there is a match then the querying agents requests the event/actions that it does not have in its repertoire.

6. If the ACM is unable to instantiate the received profile in its own knowledge base then the profile will be presented to the user for input else store this for later instantiation due to new user input or events.

7. If the agent that has been found is a mobile interactive space then if for those agents that are needed in the event/action then register with the other agents that the mis knows (repeat steps 2–7).
   else

8. Send keep alive messages to other agents with which this agent is registered.

7.2.1 State Diagram

Figure 7-3 shows the state diagram and exchange of messages, which further illustrates the negotiation that was described in pseudocode in the previous subsection. Numbering of steps mentioned below is arbitrary and only used for referencing the graphics. From the initial idle state the detection of network presence and (1) granting of network access — see also the example in Section 9.5 on page 142 — cause (2) the discovery of other entities, registration, and reception of profiles. This results in (3) an update of the internal knowledge — see also Figure 6-7 on page 99 — optionally requiring the MSK to be complemented with new knowledge, after which (5) the entities are associated. If (8) the discovered entity is a MIS, the MSK may need to be expanded with MSK from other entities in the MIS. When (6) the discovered entity is a repository, the necessary foreground & background management can be invoked. Thereafter, (7) the inference can commence, resulting in either (9) further necessary expansion of the MSK, event/actions (sent to other entities), or else (10) listen, while (13) sending a [heartbeat] pulse to keep alive the associations. (14) respond to paging requests in case the entity itself acts as a repository, (11) respond to similar discovery requests, (12)
respond to event/actions, or when (14) the device is disconnected from the network go back to the idle state (optionally caching event/actions and goals). Table I below shows the messages and argument that are passed and responses to them in greater detail.

### TABLE I XSP Messages

<table>
<thead>
<tr>
<th>Figure 7-3</th>
<th>Message</th>
<th>Arguments</th>
<th>Reply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>discover</td>
<td>from: &lt;uuid&gt;</td>
<td>found</td>
</tr>
<tr>
<td>Response</td>
<td>found</td>
<td>from: &lt;uuid&gt;, [&lt;new&gt;</td>
<td>&lt;old&gt;]</td>
</tr>
<tr>
<td>[Re]register</td>
<td>register</td>
<td>from: &lt;uuid&gt;, [&lt;new&gt; &lt;profile&gt;</td>
<td>&lt;renew&gt;]</td>
</tr>
<tr>
<td>Confirmation</td>
<td>ack</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Confirmation</td>
<td>deny</td>
<td>from: &lt;uuid&gt;, &lt;transactionid&gt;</td>
<td>-</td>
</tr>
<tr>
<td>Pulse</td>
<td>alive</td>
<td>from: &lt;uuid&gt;, &lt;expiry&gt;</td>
<td>[ack</td>
</tr>
</tbody>
</table>

**Note:** The heartbeat or pulse that is sent using `alive` messages is not only a vehicle for maintaining awareness of availability but also presence in a wider sense.

| Disconnect | bye       | -                                    | -      |
| Profile    | all       | from: <uuid>, <profile>              | -      |
| Complement | get       | from: <uuid>, [<regexp>|<template>]  | [set|deny] |

**Note:** The `get` message is used to search and access MSK in other entities. The choice and enforcement of regular expression versus a template (with regular expressions) is optional in the ACM — see further the example below.

| Fill       | set       | from: <uuid>, <instance[s]>          | [event] |

**Note:** The `set` message is used to transfer MSK to the other entity, optionally followed by an appropriate action (remote execution). Optionally, this message can transfer the entire state of the ACM (meta-contexts, mobilets, relations, goals), effectively in the teleportation of the entity.

| Event      | event     | from: <uuid>, <recObj>[<relId>[.task() | .Event[(Subject)] | - |
|           |           | <recObj> <: [ <task> | <State> ] |        |
| Fg/Bg Mgmt | kb        | from: <uuid>, init                    | [page|deny] |
| Fg/Bg Mgmt | kb        | from: <uuid>, sync <timestamp>        | -        |
| Fg/Bg Mgmt | kb        | from: <uuid>, page [<mc>|<mobilet>|relation] | [deny] |

**Note:** Cf. the `set` message but its use is restricted to transfer of MSK.

The choice of search method using either regular expressions or a template is based on heuristics in the ACM that help to confine the search space as much as possible. As the ACM knows what MSK is currently subject to reasoning, it will use the identifiers and attributes of relations, mobilets and meta-context in the immediate vicinity of the current point of reasoning in the MSK. For instance, in Figure 9-16 on page 150, Theo will prefer to send a `get` message with a `<template>` to TomAP in order to obtain the MSK relevant to the presence relation, rather than blindly querying the MIS in TomAP for any knowledge that satisfies a search pattern specified by `<regexp>`. In other situations, the ACM may need to widen the scope of its search in order to resolve a situation, for instance when in Section 9.8 on page 163, the ACM smart media player, queries the proxy in the access network for any user who happens to have any song by a particular artist, who may be close. The semantics of these concepts are embedded in the MSK itself, and neither the ACM nor XSP need to be aware of them, besides the resolution of the reasoning in the ACM and exchange of messages via XSP.
7.2.2 Service Discovery

Service Discovery follows the stage where the mobile device already has received an address and registered with a SIP-server. Service discovery uses a similar strategy to UPnP sending discovery searches and advertisements over a well-known port on the multicast channel for the local administrative domain, and unicast notification for responses to such requests. The departure from SSDP [51] in XSP is that the network topology is symmetrical; hence there are no clients and servers. Thus, unlike UPnP, where a distinction between devices and control points is made, all entities register with all other entities in a local administrative domain (within a LAN). The entities each store a local cache of service registrations that expire after a certain time. Discovery of services between wide-area networks can be accommodated in two ways. A device may be given these (foreign) registrations by another device, or a device could contact a SIP server in another domain to locate a well-known Mobile Interactive Space (MIS) agent, and delegate the discovery of services (entities) in that local administrative domain. By ordering SIP servers in a hierarchy — done by the ISP or group of local access providers (LAPs) who have a relation to this ISP — a cascaded search for all entities can be performed (for management purposes) for this ISP (with or without LAPs) to determine proper dimensioning of the network or to offer information services to third party applications providers regarding the potential and utilization of applications (i.e., data-mining). For end-points, such a register would waste vast amounts of memory, as only a limited subset of registrations of end-points would be used by an end-point. Therefore, the delegation of service discovery via a remote SIP server relieves the mobile device from this and hence is to be preferred.

7.2.3 Event Routing

A key feature of XSP is that its agent communication is based on events. The routing strategies for events differ between types of agents. Sensors do not route events, nor do repositories; only agents send events. A repository agent, which also acts as a communication space, is a communication space from an event routing point of view. Registrations involve the mutual symmetric registration of agents, and by virtue of the implicit event subscription also causes mutual subscription to events. Event subscription is implicit during registration (the service discovery phase). The following subsections, 7.2.3.1 through 7.2.3.5, deal in turn with aspects, which have great significance for the scalability and performance of event routing.

7.2.3.1 Subscriptions

During service discovery the service profile of the devices is exchanged. In the normal case, a bilateral subscription for event notifications with an expiry date is registered with each entity. Should at least one of the devices have a space attribute, then the subscription for events from the device with the space is promiscuous, meaning that notifications of received events will be sent to all registered devices. Should two spaces register with each other, then events in each space will be visible in the other and thus together they will act as one big space. As this could cause communication to explode, automatic joining of spaces is optional and enabled only by parameterization of the space attribute, specifically stating the necessary qualifications for other spaces for the join. In addition, we can use alternative strategies utilizing scoping and filtering in order to prevent excessive load of the network in dense spaces — (see Subsection 7.2.3.3).

A subscription by an object ends explicitly when the object notifies (either its neighbors or the well-known XSP entity - the Mobile Interactive Space) that it leaves (or implicitly when it times out). On the other hand the object might be prevented from sending this information because the wireless link is interrupted, in which case we allow for a default time-out value to mutually remove the subscription so that the subscriber and subscribed object may mutually decide to remove this information even though they can’t communicate, since they cannot count on being able to synchronize they must depend on connectivity being established again or both utilize the time-out value (which can be set to various values based on context) and keep-alive messages can be sent to
prevent the subscription from timing out.

7.2.3.2 Routing Strategies

We can enable different modes in agents for different event-forwarding (routing) strategies, the default being that all agents (except sensors or repository-only agents) route events to all its registered agents, which is not optimal from a performance point of view, since an agent may receive the same event from several neighbors. Other modes can be more judicious in their use of available resources (e.g., energy, bandwidth), see also [76]. Therefore, filtering of event subscriptions relative to the scope as dictated by an RDF can be applied, as entities are generally not interested in all events.

7.2.3.3 Scoping and Filtering

Scoping and filtering of events is useful for different purposes. Restricting discovery to location changes of other devices (see Subsection 8.3.1) significantly reduces the data rates generated. Similarly, we need to moderate communication needs between mobile devices in joined MISs in dense spaces, for which we can:
1. Scope routing of events, which means that events can be propagated to other spaces when the agents indicate that the event has a global scope or may only need to propagate for a limited number of hops (see Subsection 7.2.3.4).
2. Alternatively, event propagation can be filtered by the parameterization of the MIS join, where only events satisfying criteria of common interest are propagated.
3. We can restrict the registration and subscription of entities to device types that match mobilets occurring in the semantic graph of the RDF files. For instance, there is no point in trying to connect a sensor for detecting object identities directly to a printer, unless the printers MSK can be extended to use the sensor in order to enable a person to identify him/herself to the printer via the sensor.

7.2.3.4 Time to Live

Events can be tagged with a time-to-live so that they are not routed and forwarded beyond a certain number of hops. The default is two hops, in order to allow awareness between participants in two adjacent spaces that are connected, while avoiding burdening entities which are not involved.

7.2.3.5 Routing Loops

In order to avoid routing loops, events are also automatically assigned a unique object identifier, which is constructed as a tuple with two elements: the identity of originator and the universal time - thanks to GPS & NTP [64,110] it is not unreasonable to think of global time - or perhaps a uuid [10].

7.3 Object Access

7.3.1 Design Rationale

Object access may be formatted as XML in RDF-specifications similar to the use of XML service profiles in SOAP [215]. However, the generation of XML formatted text strings containing RDF specifications of the semantic graphs is an optional step, which may be necessary if universal access to these specifications and interpretation in arbitrary devices is important. Within the framework of our service architecture, internal communication in XML is optional, and generally avoided for reasons of efficiency. Therefore, an intermediate format is transferred consisting of interpretable clauses of first order predicate logic in the Prolog language. In fact, arbitrary languages can be used, as the preamble of the announce message that is used during the service discovery phase has an optional <language> field. The receiving device may accept or reject this language. If it is rejected,
an XML/RDF parser generator is expected to be used for the language in question. Should this functionality not be available (e.g., due to the limitations of the communication link or the properties of the device), the registration is rejected and the service regarded as not usable. Future development may select other intermediate languages. The impact of the characteristics of the language and its interpretation on performance, is outside of the scope of this research.

7.3.2 Requesting and Initializing Profiles

Profiles consist of a number of properties (see section Section 4.7.8). The RDF tag may refer to an RDF file on a server containing a semantic graph, which describes how the entity relates to other concepts for application management. The purpose of the RDF file is to provide external application management systems with a means to access information about the general use of the entity without needing to access and extract this information from the Mobile Service Knowledge in the Active Context Memory of an agent. The ACM may use the information in the RDF file in order to infer the intended purposes of the entity. The application management system may request the entity to send a reference to the RDF file. Furthermore, the application management system can reset the Mobile Service Knowledge to that in the RDF file, restoring its original state. In addition, the application management system can request a complete listing of the Mobile Service Knowledge in order to generate a revised RDF file with improved properties. Alternatively, the ACM can incorporate the functionality to generate the RDF. There is an obvious trade-off between burdening the mobile device with additional functionality and general compatibility. The path chosen here is to locate this functionality on servers in the infrastructure, where computational and power constraints are not an issue. The design and architecture of the application management system is outside the scope of the dissertation, see further Section 11.2 on page 177. See also Subsection 8.3.20 for a discussion of data-mining Mobile Service Knowledge.

7.3.3 Notifications

If changes occur in the external profile, a notification of this change is propagated to those entities that subscribed with this object. It should be noted that this generates a limited number of events. When entities connect to the network they may find new entities, registering with these and in turn these entities may register with it. Thus, the profiles of the connecting entity and in turn those entities it discovers will change. In principle this means retransmission of the altered profiles between parties, for example, A and B. However, this stage can be omitted as the changes in A that pertain to its relation to B are of little interest to B as it already has generated information as to its relation to A, and how A views this (same) relation is of no importance to B.

7.3.4 Extensions of Profiles

The relations of the central object or mobilet in the Mobile Service Knowledge are visible as event/actions; this object represents the ‘self’. During the course of its lifetime the use of the entity may cause a meta-context to be generated including this original ‘self’ mobilet, thus superseding this original ‘self’ with a new conglomerate self. Naturally, whether this occurs is entirely dependent on the type of application. For instance: a home CD-player will most probably not come to think of itself in any terms other than as a CD-player, but a mobile communicator, depending on what it encounters during its use, may see itself as a source of location-dependent access and communication conditions report generator, participating in a community of such devices, thus collectively optimizing their utilization of the network.

Thus, the relations to the self-mobilet or meta-context replacing the original self translate into event/actions (represented) in the external profile. The states of the self-mobilet or meta-context replacing it, translate into states (represented) in the external profile.
7.3.5 Extension of Mobile Service Knowledge

There are two different contexts in which Mobile Service Knowledge in agents is extended:
In the first case, agents discover that to interact with another agent (as specified by the RDF template and the agent profile), they need to extend their event/action repertoire and ask the other agent to forward any missing event/actions. This automated extension of the repertoire of capabilities is a key feature of XSP. Secondly, agents can apply a paging strategy [80] to transfer mobile Service knowledge to other agents, which act as repositories, using exactly the same XSP messages. The agent, which owns the mobile service knowledge, will keep a table with references to other entities (the discovery registry) corresponding to the agents that it has registered with, in order to send them keep-alive messages. The repository agent, because it has a repository property, will not publish the event/actions as Mobile Service Knowledge, but rather the Mobile Service Knowledge may be queried and retrieved using the built-in functions of repositories.

In the second case, we can specify inheritance in the RDF files that agents refer to in their type fields. This way we extend the context in which the agent specifies that it may be used. For instance: we can specify an agent to act as a meeting place (i.e. it is an abstract space agent that has suitable resources), in which people send voice messages (e.g., group voice chat). The meeting place is an abstract space with certain common resources, such as background sound from a media server, a location, a 3D model, or a list of visitors. Letting the place inherit from a digital classroom, enhances the functionality of the (mobile) interactive space. When an electronic whiteboard or a public display are placed in the room, their respective agents attach to (i.e. register with and subscribe to events from) the meeting room agent, which now inherits from the digital classroom, it sends event/action updates to participants which now enables them to share the electronic whiteboard too. This process was described in detail in Section 5.3.3 and Section 6.3.10.

7.4 Mobility

Agent mobility has been made explicit in the current version of XSP in the sense that one of the event/actions (see Figure 7-3) can be the relocation of MSK including goals and point of execution. For instance, the Personal Mobile Agent (Section 4.7 on page 65) makes use of this support in its decisions based on its context, when to transfer MSK, goals, and state of execution.

7.5 Mobile Interactive Spaces

In my paper [77], I described how a Mobile Interactive Space can be created by scaling “Smart spaces” to the size of mobile networks. Smart spaces are ordinary environments equipped with sensing systems (e.g. location, movement, visual, audio, etc.) that can perceive and react to people and conversely, by instrumenting the physical world, we enable people to influence the virtual world. The integration of the physical and virtual worlds is also referred to as mixed-reality [12]. Thus, a Mobile Interactive Space must allow any entity to engage in communication by plug and play and to respond (hopefully) intelligently. When (new) mobile resources become available and are to be used in the communication space, others must be able to locate them and interrogate them (or an intermediary) about their capabilities. In addition, there must be a naming and localization schema. There exists a close relation between, on one hand, the general application architecture of a Mobile Interactive Spaces and on the other hand specific types of user interaction, such as conversational multimedia, being an event-driven exchange of object information in parallel to isochronous multimedia.

7.5.1 Mobile Resource Awareness

Common awareness of data in Mobile Interactive Spaces is created through the accumulation of Mobile Resource Knowledge and sending event updates to registered entities within this space. An example of this concept and design choices is available in [155]. The agent which hosts the Mobile Interactive space, which stores shared knowledge as an associative tuple-space [99].

In my paper [77], I described how a Mobile Interactive Space can be created by scaling “Smart spaces” to the size of mobile networks. Smart spaces are ordinary environments equipped with sensing systems (e.g. location, movement, visual, audio, etc.) that can perceive and react to people and conversely, by instrumenting the physical world, we enable people to influence the virtual world. The integration of the physical and virtual worlds is also referred to as mixed-reality [12]. Thus, a Mobile Interactive Space must allow any entity to engage in communication by plug and play and to respond (hopefully) intelligently. When (new) mobile resources become available and are to be used in the communication space, others must be able to locate them and interrogate them (or an intermediary) about their capabilities. In addition, there must be a naming and localization schema. There exists a close relation between, on one hand, the general application architecture of a Mobile Interactive Spaces and on the other hand specific types of user interaction, such as conversational multimedia, being an event-driven exchange of object information in parallel to isochronous multimedia.

7.5.1 Mobile Resource Awareness

Common awareness of data in Mobile Interactive Spaces is created through the accumulation of Mobile Resource Knowledge and sending event updates to registered entities within this space. An example of this concept and design choices is available in [155]. The agent which hosts the Mobile Interactive space, which stores shared knowledge as an associative tuple-space [99].
7.5.2 Dissemination Algorithm

We can also envisage the deliberate dissemination of mobile service knowledge. In his dissertation [103], George Liu shows how location information can be distributed in optimal ways in mobile networks, thus providing important input concerning effective ways to disseminate mobile service knowledge, which can be combined efficient strategies for locating application data across service discovery domains [30] or dissemination [93]. New scenarios cause us to re-examine these strategies as people, mobile devices, and resources may meet and exchange knowledge without necessarily be connected to a network. Thus the requirements on the dissemination algorithms have changed in character, due to the increased dynamics of the communication.

These issues are especially relevant to the event routing that was discussed in the previous section, where the result was sent to the neighbors with a time-to-live attribute attached to it. At present no quantitative measurements are available regarding the performance of such a strategy. However, based on experience from [81,83] we expect that this strategy probably will work well.

A MIS offers awareness of its registered entities (see further the example in Subsection 7.5.1). Thus when entities appear and leave, these changes will be visible to other entities present in the MIS using the presence mechanism in XSP. Furthermore, when entities acquire new capabilities these capabilities are in effect new services, and announcements about these services will be propagated between entities.

7.5.3 Access Control

Access control and security are important for various reasons. We need to be able to create groups of trusted parties. MISs must be prevented, using information from access-control and security profiles, from forwarding MSK to untrusted parties. Users may want to prevent involuntary data-mining their ACMs by others, such as service providers. Therefore, ACMs can include profiles (for users as well as for other entities) with an access control structure, specifically stating under what conditions identities or classes of other parties may access parts or all of the MSK in the ACM. The design and exchange of such a profile is outside the scope of the dissertation, and it suffices to say that such an access control record and security profile can be added to the ACM and used by XSP in order to implement the necessary restrictions thus ensuring the user’s privacy.

7.6 Negotiation

When an entity discovers another entity and registers with it, the ontology of this device is transmitted to the other and merged. Clearly, their merging requires the receiving entity to have an entry in its ontology for the discovered object; otherwise a hypothetical ontology needs to be synthesized. The synthesizing of the ontology is outside the scope of this research, but clearly, Case-Based Reasoning (see [111]) enables a device to accumulate cases and synthesize & select successful models.

7.6.1 Collective Behavior

Evaluation of the MSK in the ACM determines the adaptation of the communication by the agent. Their adaptation following events may involve several entities and their respective Active Contexts, as these events may trigger rules in several ACMs, potentially leading to other events being generated, leading to other rules being triggered, etc., until this process stops. Consequently, we may even regard this collective behavior of agents (and their ACMs) as the application itself. For example, a user entering a room with a networked public display, microphone, and loudspeakers, may thus collectively cause the user to make a video call home, should such a request be pending (stored as rules in the user’s ACM), when the sudden availability of the correct resources causes these rules to trigger.
7.6.2 Monitoring Behavior

As, XSP allows the retrieval of MSK from the ACM, service providers could request and mine the ACMs of users, mobile devices, or resources, for data pertaining to their particular service. Knowledge exchange and retrieval, facilitated by XSP, is used for the automatic distribution of mobile service knowledge between the mobile personal agent and the stationary personal agent in order to support off-line behavior of users. See also Subsection 7.5.3 for restricting access and maintaining privacy.

7.6.3 Exchanging Behavior

Using XSP, any entity can export adaptive behavior comprising Mobile Service Knowledge to Active Contexts of other entities and thus allow these other entities to benefit from their accumulated knowledge. Consider the following examples for end-users, regarding the two types of adaptation (Section 6.1):

- The user has inherited knowledge from others of locations with lower throughput and because they expect to pass through this location they use this knowledge to decide to pre-load required content while in other locations with higher throughput.
- The user has inherited knowledge about locations with public resources. Other users may have attached rules with additional information about the resource to this knowledge (e.g., information explaining how to use the resource); thus when a user enters this location they will already know of the resource and how to use it.

Exchange of knowledge between users may be done consciously in a type of dialog, but could very well occur automatically as the result of membership in a mobile user group (implying that members share information and resources). Forming mobile user groups opens up the possibility to build taxonomies or directed acyclic graphs (DAGs) of users, where knowledge is automatically propagated to other users who have inheritance attributes in their profiles from others. See also Subsection 7.5.3 for restricting access and maintaining privacy.

Likewise, the usability of artifacts (that is, a networked resource, e.g., a coffee machine, printer server, digital camera, or a public display) will benefit from being able to exchange service behavior as well: the artifact may store knowledge of preferences of users or subscriptions to notifications by users. When introducing a new device, it should inherit common knowledge from similar devices (e.g., when plugging in a new coffee machine, it should be made aware that I prefer black coffee, by getting this information from MSK in my ACM created by other coffee machines I have used).

Obviously, virtual (intelligent) objects, lacking any physical existence, share all these same properties regarding the interchange of knowledge between object instances (even if the existence is virtual).

Figure 7-4. Merging Ontologies
7.7 Summary and conclusions concerning an Extensible Service Protocol

This chapter presented the eXtensible Service Protocol (XSP), thus concluding the presentation of the three key components of the open service architecture, the other two being Mobile Service Knowledge (MSK) for specification of behavior and the Active Context Memory (ACM) for reasoning about this specified behavior. XSP is simple but powerful protocol, which, when combined with MSK and the ACM enables dynamic negotiation of service behavior between end-points in order to create context-aware applications. XSP is agnostic about the service behavior; this allows us to extend MSK and the content of ACMs, without changing XSP. XSP enables ACMs in entities to exchange and negotiate Mobile Service Knowledge and thus achieve an end-to-end negotiation of services, while managing and extending MSK. XSP makes very few assumptions about the capabilities of communicating agents and therefore the requirements on a-priori (shared) knowledge are limited to the concepts of the architecture, not the (knowledge or the concepts) of the applications themselves. XSP is event-based and sends messages between agents on a peer-to-peer basis. An agent may, but is not required to run in server mode, depending on the type of agent. XSP recognizes a few, but important, classes of special purpose agents: sensors (emitters of events with data from measurements), Mobile Interactive Spaces (MISs) for creating smart spaces, and Repository agents for the foreground/background processing, as well as data-mining of MSK. In comparison to UPnP, XSP has properties for discovery, registration, and eventing similar to UPnP. However, while UPnP does not clearly tie services to the objects for locating and controlling these services (a control point), this mapping is central to XSP. Therefore, in combination with MSK, XSP enables the ACMs in agents to negotiate the semantics of communication and event/actions, going beyond what has previously been possible.

The design and management of access control and security profiles in agents remains open issue, which need further investigation. The next chapter investigates the network aspects and requirements of the service architecture and its components.
8 NETWORK ASPECTS AND REQUIREMENTS

This chapter addresses the issues of how of to distribute functionality of the Service Architecture Framework and what trade offs can be made regarding different scenarios. For this purpose we shall consider different scenarios for how the Service Architecture Framework and its components can be applied.

8.1 Introduction

Not only is it important to show what applications have been enabled in a mobile computing and communication infrastructure, but we must also be concerned with such issues such as network economics, scalability, management, and step-wise introduction. These requirements are important factors for these applications and solutions that we have discussed above to be successful on the scale of today’s networks for fixed and mobile telephony. These areas are addressed for existing and new operators using the reference models in Figure 2-4 on page 28 and Figure 9-1, when we discuss in the sections below how the open service architecture that was presented and explained in previous chapters can be applied.

8.2 Impact on Service Delivery Frameworks

This section investigates ways to adapt existing telecom service architectures and interact with the service architecture that has been described in this dissertation:

8.2.1 (Advanced) Intelligent Networks

The IN-services that users of existing telephony networks use are not (necessarily) affected. However, service providers may offer end-users redirection of their Universal Personal Numbers to their new communication identities on the Internet, thereby offering to include their existing services in their new application profile. This is easily achieved via existing CCS7-gateways between telephony networks and the Internet. This also allows events that are handled by the

![Figure 8-1. Reference Model](image-url)
communication identities on the Internet to trigger a notification to the Intelligent Network, in (the rare) case that no Internet access is available, but a telephone is. Much more importantly, it allows application providers to use the information that they already have available in Intelligent Networks to their advantage in these new solutions. For instance, the customer administration databases together with the information in the IN nodes can now be a real asset when combined with interactive VoIP-enabled e-commerce on the Internet.

8.2.2 CTI

Providers of Call Center solutions may in a similar fashion map events to actions. As Call Centers utilize computers in a local area network in which the data for transactions and customer administration are processed, Active Context Memories can be attached to react to customer behavior. Thus, customers who are interested in a certain product can be automatically routed to the proper call center for further assistance. Unlike programmed solutions, the software structure of the Active Context Memory allows knowledge that relates to exactly this destination (or more generally this context) to be inherited (and thus automatically updated when changes occur). However, with a very moderate investment (paid by savings in infrastructure) and very little effort, CTI solutions can incorporate VoIP in their user interfaces, in which case interfacing of events between the existing solutions and the service architecture that has been presented in this dissertation can be eliminated. The result is a Web based Contact Center that is indeed very attractive to the new application providers described in the next section.

8.2.3 GSM

GSM implements a slightly modified service architecture, but what was said about IN, also applies to GSM-customers. Due to the fact that mobile phones also are capable of receiving and sending SMS-messages, it is possible to increase the interaction with the user’s communication identity on the Internet. This can, for instance, be achieved by responding to short messages sent by SMS or send commands via an SMS Server and gateway (transforming it to e-mail) back to the Internet. Since the movement of the user generates events in the GSM network, location-based interactions are possible. The network provider could send location information via a gateway from the base stations to electronic identities on the Internet. These service providers can offer customers hosting of their home pages, electronic identities, and free storage of content. This allows these service providers to send pieces of knowledge to the Active Context Memories (see Chapter 6) that react to events such as the mobile being in a certain location, upon which a commercial advertisement could be sent. Thus the GSM operators have started selling their customer base to those who have products to sell to them. Further advanced scenarios are possible if the operator maintains Active Contexts attached to user locations, which in turn send notifications to mobile users (e.g., a location dependent information system for smart dynamic traffic control, where the user can now react to this information in the real world).

8.2.4 3G IP-MM

The standardization regarding a 3GPP IP MultiMedia Subsystem in R4 and R5 of 3G IP-MM [204] refers to a “service network”, so neither services are networked nor separate networks for services are created. In fact, the so-called “service network” is a business model dependent service delivery framework; i.e., it specifies a set of interoperable components and network nodes, such as proxies, media gateways, and content servers in the mobile operator’s network, which enable a mobile operator to host services for a multimedia application provider. This framework is made business model dependent by coupling AAA to charging, invocation, and delivery of specific services, preferably via proxies in the operator’s access network. The proxy nodes are then used to collect information about sessions and services and send this data to a billing center. Our open service architecture is also applicable to this business dependent service delivery model, but the requirement
to have a third entity (the operator’s node representing AAA, charging, invocation, and delivery of services) blocks attempts by the end-points to spontaneously introduce, negotiate, and exchange new capabilities or services. We can reduce these limitations as follows:

1. Decouple AAA (i.e., obtaining network access) and its associated charging from functions that do the necessary user accounting, charging, invocation, and delivery of specific services. This enables users to access the services via other network operators’ networks (and perhaps pay arbitrage for being able to access this network operator’s network).

2. In addition to the previous point, decouple the user account (and personal service profile - i.e., what corresponds to the mobile agent in the open service architecture) from the support functions for charging, invocation, and delivery of specific services. This enables users who are not a customer of this network operator (where the services reside) to access the services (and perhaps pay for being able to use these services).

Hence, we have subdivided the previous-one-stop-shopping role of operator into an access provider (offering access to the Internet), a network operator (providing support functions for hosting and delivering services), and a service provider. The remaining limitation is user accounting coupled to charging, invocation, and delivery of specific services.

This remaining limitation, which is due to the business model of existing operators and cemented in their existing infrastructure restricts the services available, as users are only allowed to extend their communication models spontaneously within the bounds that have been dictated by the network operator and service provider, who in many cases will be one and the same.

The remaining requirement to couple user accounting to charging, invocation, and delivery of specific services demands that (1) support for XSP, (2) the mobile agent & ACM, and (3) support for management of MSK are integrated in the network operator’s support functions for hosting and delivering services. Thus, the following requirements can be derived for how the APIs and layers in the 3GPP IP Multimedia Subsystem can evolve [204] to accommodate and be interoperable with my more general service architecture. These requirements outline ways to move forward for existing operators who move into πG (see Section 3.7) via GSM (e.g. Telia) or directly (e.g. NTT DoCoMo). This also opens up the architecture for new application service providers considered in Section 8.2.5:

**Figure 8-2. Network Providers**

- **XSP Support**
  - Aside from the end-points, an XSP-capable proxy is needed in the network operator’s network, since the network operator is required to monitor and support the negotiation of communication for the service provider.

- **Mobile Agent**
  - The User Profile must be allowed to roam from the "home network" to the "visited network" (Figure 8-2). In addition, this user profile must be extended with a SIP-UA (stack), XSP, and an ACM.

- **MSK Management**
  - A repository for MSK should be available, to which infrequently used knowledge may be moved for archiving or backup purposes, and subsequent retrieval or deletion.

**8.2.5 Network Operators**

A Network Operator can be anyone from an extended telecom infrastructure to power distribution...
network as long as they provide a packet network point of presence as an access provider. In real world scenarios, we will see different network operators providing access to their networks. In Subsection 2.6.5, I even stated that anyone could be a potential network operator by allowing users to get access; provided support for authentication and charging for used bandwidth is available (and desired). Thus users may expect to move between very many wireless networks. In Figure 8-2 a user who has a home location in network A moves to network B. Both in the case of roaming or a handover, the user is re-registered in the SIP-server of network B. Hence, SIP invites will be forwarded to the new location. If some intelligent action is required the SIP server in the new location can forward the invitation to the agent in the mobile. However, if the mobile goes off-line, the invitation times out, unless the SIP agent is a SIP-proxy that uses a location server that can poll the mobile and when the mobile has gone off-line it redirects the invitation to the master copy of the agent.

Moving forward to a business model independent service delivery framework (in contrast to the business model dependent service delivery framework in R4 and R5 of 3G IP-MM) In Section 9.5 I described how the service architecture is applied to create such a business model independent service delivery framework. The service architecture framework can thus be applied in order to enable entirely “open” communication where anyone can buy any service from anybody else - thus completely separating the roles that a single operator has in existing telecom networks. In the following subsections I address the requirements and implications for a number of (potentially) different parties.

8.2.6 Application Providers

An Application Provider can be anyone from a private person with a mobile device to a major content deliverer (e.g., CNN). Not only can voice services be integrated into web based services (see [69,71] or VoiceXML [216]), but in particular, application providers are able to enable their services to react to any event caused by customer behavior, and use it to their advantage to maximize their ability to meet a customer need (assuming the user is a customer of their service). In most cases, customers will be asked to register {when wouldn’t a customer register?}, in which case such a new application provider activates a customer profile that can now be enhanced with a personal agent and active context memory, all of which can be downloaded on-line in the background for future interaction. In the following sessions, the application provider might send pieces of knowledge that will increase the user’s awareness of certain products depending on the user’s preferences and behavior when visiting the on-line or off-line services. However, users must be able to filter their events and determine who is allowed to receive them; see Subsection 7.5.3 for more information regarding scoping and filtering of events. Another example, vendors of on-line kitchen equipment1 find new possibilities to tailor the behavior of their devices to the individual preferences of users or groups of users (e.g. for families with an electronic blackboard metaphor notifying who is in charge of today’s dinner along with shopping list hints based on the status of the equipment necessary for its preparation). Similarly, vendors of on-line toys can exploit this to personalize the interaction with kids and invent personal toys.

Furthermore, new application providers will be able to simulate and predict the behavior or success of their services prior to general release among their customer base. For instance, a simulator or local network can be created to host agents for all the entities (e.g., proxies, Mobile Interactive Spaces, users) and these agents can be instantiated with the Mobile Service Knowledge (MSK) comprising the semantic models for the user, the MSK for the application to be sent, preloaded into the ACNs of these entities. The user agents can be caused to send events to the application server. Thus, the application provider can observe how the application behaves in this simulated environment from a user perspective, while also seeing what demands the application places on nodes that participate (e.g., a proxy). Hence, the Active Context Memory enables an application provider to pre-design encapsulated knowledge that, when loaded in the Active Context Memory, will instantiate specific

1. E.g., http://www.electrolux.com/screenfridge/
behavior due to certain events. This is the application and hence allows Application Providers to simulate and predict how applications will behave even though the service might not yet formally exist. The importance of this is that it enables the Application Provider to know in advance the behavior and performance of the application, both in terms of whether the interaction will be effective from a user perspective, but also to determine if special precautions need to be taken (e.g., dimensioning of network nodes) in order for the performance to acceptable and the service useful. In addition, the functionality allows them to incorporate reporting functionality in services for subsequent data mining, all to their advantage (again, this assumes the customer has approved of this “service”). As mentioned earlier, users must be guaranteed their privacy; see Subsection 7.5.3 for more information regarding scoping and filtering of events, preventing involuntary participation in the acquisition of MSK data by an application provider. However, we believe that there will be users who will provide some of their MSK to application providers, either to receive a monetary or service reward or because of the improved behavior their devices will exhibit.

### 8.2.7 Credential and Access Providers

A Credential Provider is a trusted party confirming the credentials of communicating entities, for instance to acquire network access. Hence, not only can gaining network access be done anonymously but so can gaining access to a service. A credential provider is not part of the architecture, but as shown in Section 9.5, the architecture enables access providers in combination with credential providers to develop new types of communication and business models.

An Access Provider can be anyone from a private person with a WLAN extension to fixed Internet to a 3G network operator. Thus the implications for AAA are that XSP enables direct negotiations enabling anyone to become an access provider who may use a third party (such as a credit card company) for AAA. AAA may not be necessary for some in the access networks.

### 8.2.8 Meta-Application Providers

This section examines the necessary requirements and properties of solutions for a new class of service providers, so called “Meta-Application Providers”, which utilize the service architecture and its components that were presented in earlier chapters.

The ACM maintains MSK that describes a specific context, such as a meeting, a play situation (with a toy), or a copying context. This capability enables meta-application providers to design skeletons.

---

2. Note the access or service does not necessarily need to know whom the customer is, only that the credential provider has vouched for them and that the credential provider guarantees payment.
and behaviors. When users enter such a specific context, they will instantiate unknown variables and extend this knowledge as a result to their behavior. Extension is accomplished by means of aggregation, as shown in Figure 8-4 and described in earlier chapters. Meta-contexts contain slots with references to the instances themselves (a Location, a Public Display) or another Meta-context (a Group), which in turn has slots with references to instances of users (Person A, Person B). Meta-application providers are thus providers, which unlike application providers do not necessarily have a continuous logical presence as application providers normally would be expected to have (e.g., a mobile portal). Meta-application providers are those that can be reached through such a portal, but not exclusively, so that any entity can be the carrier of such a meta-application. Hence meta-application providers do not require the existence of a portal.

In Subsection 6.3.11, Section 6.5, and Subsection 8.3.5, I discussed some aspects regarding backup and (re-) synchronization strategies of the Active Context Memory with respect to off-line behavior. Hence, these repositories for MSK can also be used for maintaining, managing, and providing meta-applications based on transferring MSK via XSP.

8.2.9 Examples of Uses

As illustrated in Figure 8-3, a meta-application provider could be an entity (server) on the Internet who provides meta-applications that plug into the MSK of other entities or mobile users (e.g., a@b.c) having a meta application that they can and do decide to share (e.g., a chat-board for a mobile journalist who will reflect instant messages that come in and comment on the live multicast reports that he or she is sending).

Another example is a package of MSK that when it is plugged in reacts to certain events and instantiates an application. For instance, knowledge can be added to the Active Context Memory of the Public Display, which enables it to respond intelligently to information that is sent to it. For example, it may ask the sender if a hard copy is needed or if the information should be forwarded to a Group. This group is a meta context and is able to handle the sent information and distribute it appropriately among its members (see Figure 9-4 and Figure 9-5). In addition, Person A might be a member of the Group or made a member of an ad hoc Group that the Public Display administers, in which case user communication instantiates group communication (e.g., Chat).

8.3 Networking Aspects

This section discusses networking aspects for the provisioning of services based on the service architecture and its components. The following subsections examine the requirements that need to be considered for building systems based on the service architecture and components with respect to networking situations. However, it is important to note that we are examining the results from a research project and that more aspects need to be addressed before such a system might be rolled out for mass use, these aspects have to do with the design of target systems (e.g., GPRS, 3G, etc.), rather than the principle aspects of networking which are studied below.

8.3.1 Presence and Location Support

As we shall see in Section 8.2.4 and also described in [78], we can co-locate a SIP-server and
well-known XSP entity forming a Mobile Interactive Space (MIS) with an access point. XSP allows entities to discover each other just using multicast channels (without the assistance of a 3rd party). In addition, it also provides support for an alternative strategy, where a well-known XSP entity, such as a Mobile Interactive Space, co-located with a SIP server, offers information about the presence and location of entities that it has discovered. The SIP-server uses the Mobile Interactive Space as a location server, in line with the SIP RFC 2543 [59]. The advantage of this approach will be evident in the examples using cellular networks below.

8.3.1.1 Example

Assuming the worst case of a power outage of a GPRS radio cell with M mobiles, restoring awareness and refreshing all caches, the number of discovery requests D between mobiles equals 1 and the number of reply notifications R between mobiles equals D, the average number of bytes per notification A bytes, and the time allowed for restoring this information T. In an extreme example, let: M = 1000 (mobiles), A = 512 (bytes, cf. the size of discovery requests in [51]), and T = 60 (seconds). Then the required network bandwidth, \( B_{\text{network}} \), equals the product of the combined discovery & reply message between all mobiles divided by the restoration time:

\[
B_{\text{network}} = \frac{(M) \cdot (D + R) \cdot A}{T}
\]

In this example, the worst-case generated data rate in the network is \( B_{\text{network}} = 499500 \times 2 \times 512 / 60 = 8.52 \text{ Mbps} \). This level of traffic can easily be accommodated for with fast Ethernet backbones, provided we disallow network wide discovery. Hence, the worst-case generated data rate over the GPRS wireless link, \( B_{\text{link}} \), is calculated as the product of a combined discovery & reply message from the mobile to all other mobiles, divided by the restoration time:

\[
B_{\text{link}} = \frac{(M - 1) \cdot (D + R) \cdot A}{T}
\]

Given the above values for M, D, R, A, and T, \( B_{\text{link}} = 17.05 \text{ Kbps during 60 seconds} \). This load on the link cannot always easily be accommodated over the wireless link, as GPRS offers data rates of 13 kbps and above. Moreover, it cannot be regarded as optimal when a minute is needed for all mobiles to recover full awareness of each others presence, as this does not include the exchange of Mobile Service Knowledge.

8.3.1.2 Improvements

However, we can do much better than blindly trying to discover and re-register with every device, due to the observation that we only need to consider changes. Mobile devices that have moved in the network can be reached at their new location via Mobile-IP, and sessions can simply be reestablished when the mobile device registers with another SIP server. The mobile device’s registration may time out, or it may choose to leave the Mobile Interactive Space (MIS). Thus, when mobile devices need to refresh their caches upon reconnecting with the network, they can get a list of location changes that have occurred since the disruption of the communication, and that match this information with their caches, and hence need only register with entries that have appeared, and remove all others. The computational penalty for reducing the network traffic is small. However, the gain in network efficiency is significant:

When reconsidering the previous case, after recovery of the link, the mobile will send an update request to the MIS with its own status (e.g., the time when it received its last update) and obtain a list of mobile devices that have immigrated (I) or emigrated (E) respectively in the meantime. Emigrants
are simply removed from the cache and hence the mobile need only register with the immigrants. Assume \( I = E = 250 \), corresponding to a quarter of the population having been replaced by new individuals, and each mobile (re)registers with the MIS either to update its status or to join the MIS, requiring only exchanges between immigrants and indigenous mobile devices \( (M^I) \). Then the new load on the network \( B^i_{\text{network}} \) and the link \( B^i_{\text{link}} \) are calculated as follows:

\[
B^i_{\text{network}} = \left( \binom{I}{2} + M^I \cdot (I + 1) \right) \cdot (D + R) \cdot A \frac{1}{T}
\]

\[
B^i_{\text{link}} = \frac{(I + 1) \cdot (D + R) \cdot A}{T}
\]

Then during a 60 second period, \( B^i_{\text{network}} = 0.53 \text{ Mbps} \), which is less than an order of magnitude of the previous value and \( B^i_{\text{link}} \) is reduced proportionally to 4.28 Kbps.

### 8.3.2 Feature Interaction

In this dissertation, I presented a service architecture and components, which enable the direct network-wide exchange and negotiation between end-points (i.e., not requiring a third entity to intervene) of Mobile Service Knowledge via the eXtensible Service Protocol (XSP), and thus extend their local formal models, which together with reasoning in the Active Context Memory (ACM) comprises goal-directed behavior. Feature interaction (see Section 3.12 for related work in this area) on a local level may occur on three different levels, i.e., interaction or collisions concerning the structure, semantics, or intentions of MSK. I will cover these three levels in order:

#### 8.3.2.1 (Local) Structural Integrity

Feature interaction with respect to the structure may occur locally, and potentially cause collisions, due to the fact that MSK comprises partially ordered sets of quanta of service behavior (relations, mobilets, meta-contexts) and merging of MSK may not be successful or even possible. Previous results for verification, detection, prevention, or resolving feature interaction on this structural (logical) level in real time (see Section 3.12) can be applied to the reasoning support and incorporated in the ACMs, thus ensuring the structural integrity of the of the MSK, even on the level of aggregating packages. Addition of new MSK to the ACM (e.g., Figure 9-18 on page 151) will (irrespective of whether it is contained in a package or not) causes the integrity checker in the ACM (Subsection 6.3.7) to perform a logical verification of type matches of mobilets, meta-contexts, and relations, along with the examination of the topology of the resulting graph. Matching types of relations, meta-contexts, and mobilets, cause these to be resolved. Mismatches cause the rejection of isolated groups (unresolved relations) or the entire cluster of MSK.

#### 8.3.2.2 (Local) Semantic Integrity

To be able to deal with feature interaction on a semantic level, we need to include a-priori knowledge of key service concepts in the ACM, such as in RKRL [111], or as in other approaches by including procedural knowledge of these concepts [160], which means this knowledge is explicit in the sense that the support is built to recognize these concepts, and is able to act accordingly. The explicit default incorporation of a-priori known service concept is also present in other approaches involving negotiations of contracts between agents [4,54,116]. The approach presented in this dissertation differs from these previous ones in that it completely lacks a-priori knowledge, both explicitly, implicitly, and as a default mode of operation. The theory behind this, is that MSK is a represents the observed objects in the communication space and their relations (and therefore behavior) — see Section 5.1 on page 79. This observation has consequences for how we can deal with
feature interaction on a semantic level, as MSK specifies the semantics. Therefore, ensuring structural integrity of MSK, also ensures semantic integrity.

8.3.2.3 (Local) Goal Integrity

Finally, I examine goal integrity, but then we must first define what constitutes goal integrity. When MSK is merged, goal integrity ensures that the a-posteriori conditions are preserved when we add the new goals to the set of existing goals. We can guarantee a moderate degree of goal integrity by inspecting the goals, and examine whether the new goals will generate values and actions, similar or which interfere with the old ones. Therefore, we also need to inspect the semantics of the mobilets (their properties and the relations between them). For instance, using the examples in Section 9.5, we may receive a package of MSK, which enables us to establish communication with a large ISP who uses own methods for AAA. If (in Figure 9-5 on page 142), theo moves to one of homerun wlanAPs afterwards then the new MSK will add similar methods for acquiring network access. The goal of theo is (in the first stage) simply to gain network access, and therefore a naive analysis of the goals would lead us to believe there is a feature interaction problem, but as MSK registers the semantics (i.e., relations with the objects that these goals pertain to), we can easily determine that goal integrity is preserved.

8.3.2.4 Distributed Negotiations

To be able to understand the problems involved, consider the following example: assume the application to be well-defined with limited functionality (most likely because it is a familiar service) such as a teleconference employing Internet telephony (see Figure 8-4 on page 120), then one of the parties must act as moderator sending the (closed) model of the application to other parties for acceptance (in which case the model is executed as is) or rejected. We have already dealt with feature interaction on a local level, so we are able to deal with the MSK model of the meeting and accept, reject, or more importantly a counter proposal. In this case feature interaction can occur because of the design of the meeting model, when the relations in this model does not address how participating can reach an agreed model for the communication. It is therefore probable that such well defined applications are the result of a design effort and have been subject to validation and simulation prior to their release, rather than as a meta-context the synthesized result of an ad-hoc multi-party meeting. The presumed designer will therefore have modelled the central meta-context meeting object to act as the moderator assuring participating parties will agree in a determinate way. Feature interaction in the (closed) model is then detected and prevented by including the strategies from e.g., [4,68]. Therefore, I envisage the design of application-specific MSK packages, which encapsulates this behavior. Strategies for the design, management, and distribution of packages are addressed in the subsections below.

8.3.2.5 Emergent Behavior

If the application is not well-defined with open ended-functionality or use of resources (in order to change mode of operation relative to the context of the user or the communication context), the concept of feature interaction must be reinterpreted to apply to the evolving purposes of the MSK in the ACM on an abstract level, as it involves reasoning about potentially conflicting purposes rather than problems in ordering of quanta of functionality to produce an end-state given a range of initial states. Hence, the ACM is able to reason about MSK in order to discern which action is reasonable given the context of the user or communication in order to satisfy the purpose specified in the model. Whether the response is reasonable is measured in terms of goal satisfaction. The important conclusion from this is that when new knowledge is added to the ACM, the ACM must be able to identify conflicting goals and either resolve this conflict or bring it to the attention to the user.

For example, if the application is to download and cache multimedia content via heterogeneous access networks, then the sudden detection of a nearby hotspot (WLAN access point) to which the
user is moving should support the current download via wide-area access. Given the stated user preferences for price/performance and delivery deadlines, the ACM will drop the wide-area access after AAA negotiations with the local hotspot have been completed. If user A previously has stated that he/she also is interested in the presence of person B in a trusted user group. Then if the appearance of B and communication between A and B interferes with the download process, the ACM should defer download of multimedia and instead invite B while getting the attention of the user.

8.3.2.6 Conclusions concerning feature interaction

This section examined in what ways the problem of feature action manifests itself in this new model of communication and the relative success of strategies for eliminating these problems. In a broader perspective we conclude that for services that are well defined (e.g., multimedia conferencing) the inherent semantics of MSK enable the ACM to maintain the integrity of the application, but we also noted that the combination of MSK and the reasoning in the ACM supports the creation of emergent behavior (i.e., due to MSK that has been synthesized out of accumulated events). For these classes of applications we need to reinterpret the concept of feature interaction, and conclude that the ACM can maintain an integrity of purpose, provided we add to it domain specific service knowledge for the ACM to be able to determine whether certain actions are reasonable.

8.3.3 Network vs. Client Considerations

A current trend in computing and communication is to develop thin clients, meaning the requirements on software and operating system that is stored in the client is kept to a minimum. This has been shown to work for some types of applications with wired access. Even in these cases users may find themselves waiting for the end-station to become functional.

In wireless communication we cannot assume the network will always be present and if it is that it provides amounts of bandwidth sufficient to transport a major part of the software that has to run in the client, when the client is powered up. Therefore, we will have to assume that mobile units will come with an operating system and some client software, such that the mobile will instantly become functional when powered up. Furthermore, when wireless access to packet networks is available this software enables us to download content and functional components to the mobile perhaps even as a background action. It also allows us to do so intelligently by exploiting knowledge in the active context memory.

However, this scenario presumes that we identify ourselves uniquely with one mobile device making it our personal device. I would rather see that we are able to pick up any handset and turn it into our own. The solution is to store the personal agents and active context memories in the network, in the sense that these are stored on-line on agent servers with personal profiles (consisting of a personal agent and active context memory). These can be downloaded to personalize a client device.

8.3.4 Mobile Agents

The personal agent and active context memory of a user are implemented as mobile agents, such that once the mobile is turned on and after identification of the user, these mobile agents are allowed to migrate from the agent server to the mobile (in Figure 8-2). Identification of the user can be done as a logon procedure on a web page accessed via SSL. Should the network not be present then off-line copies of the agents are available that have been stored in the mobile, provided the device has been used by the user previously.

8.3.5 Synchronization

The obvious and straightforward strategy is to assume that the copies of the agents in the mobile always synchronize with the master copies in the agent server of the home location. If the network becomes unavailable any interactions with other agents on the agent server in the home location are
buffered in message queues as are the interactions in the off-line agents in the mobile. The on-line agents on the agent server need not necessarily be aware of the mobile being off-line temporarily. For instance, if a call is received, it can still be processed by the master copies of the personal agent and active context memory and the call routed to a voice server. On the other hand when user intervention is required, such as bidding in a web auction to which the user has subscribed, such an interaction is queued.

The question arises as to where to move the agents when the user moves. If an agent server is available in the new location, then the mobile agents can migrate to the agent server that is co-located with the SIP-server (in Figure 8-2). If no agent server is available then the SIP-server must have proxy capabilities for querying a location server that is able to tell where the master copies of the agents are when the mobile goes off-line. Clearly, this introduces additional degrees of freedom, but also greater complexity.

If the user moves multiple times from the home location (H) to a series of new locations (L1, L2,…Ln-1), then the forward reference to the current location (Li) in the SIP server at the home location has to be updated, since the URL referring to that server is the one by which the user is globally known. In case a new location (Lk) does not provide an agent server, but the previous one did, then the agents migrate back to the home location. In case tracking of user movement is an important goal, then all that this requires is updating the profiles of users. It does not require intermediate locations to maintain and synchronize user profiles.

8.3.6 Dissemination

For instance, if users roam around in an urban area with ubiquitous anonymous access with greatly varying bandwidth, latency, packet loss, with agent-enabled MP3 players, then clearly these users will benefit from sharing information about the relative quality of service in different access networks and at different locations; along with information about how the mobile who visited this location earlier was able to successfully deal with the network access situation. In this case, users must be judicious about whom to send this information to. Clearly, sending everything, continuously to everyone else defeats the purpose of this approach, because it will drown the network with updates; while consuming a significant portion of the available bandwidth for transmitting these events, which may not be relevant.

Similar to the recovery of the discovery information we could assume updates each minute resulting in a data rate in the backbone (B_{backbone}) for 1000 mobiles (M), 1000 replies (R), an average size of 512 bytes per reply (A) resulting in B_{backbone} = M * R * A / 60 = 8.5 Mb/s and B_{access} = 8.5 Kb/s, which is totally unacceptable. Alternatively, the Mobile Interactive Space agents, which are co-located with the SIP servers near the access point could aggregate this information. Hence, mobiles would be able to request information about communication conditions on demand, thus reducing the bandwidth required for transmitting updates down to a fraction of what would be required otherwise.

For the information about communication conditions, this makes sense since the information regarding data-rates is related to local locations relative to the specific access point. We may even group access points (e.g., for a WLAN segment) and set the local administrative domain to encompass these access points (e.g., for a suburb in a city). In such a case, the SIP server will not be co-located with the MIS (i.e., the well-known XSP-object Mobile Interactive Space). It is less likely that there will be a MIS per access point for local area access (as this increases the cost of the otherwise inexpensive access points), but most likely there will be only one SIP-server for such a service area (see Figure 8-6).

Users tom@fatburen.org and theo@it.kth.se are both registered with the Mobile Interactive Space at Bågen and use its Media Server. Their registrations may have occurred due to finding Bågen’s MIS on the world wide web, e.g., published on Bågen’s home page as a SIP URL or these users may have been directed to Bågen’s MIS indirectly, e.g., when negotiating network access with
a local access provider (Section 9.5 on page 142). Simultaneously, they are registered with a MIS for the access network that they are visiting. The MIS has been registered as a well-known XSP-agent in the DNS’s MX-records, in parallel to registering a well-known Mail Transfer Agent (MTA). When the mobile moves and enters a new Mobile Interactive Space (MIS), the MIS announces a reporting service for communication conditions in the local access network to these users. As these users are both using the same Bågen MIS, knowledge about access network condition reporting functions can be exchanged between them, as well as the exchange of data regarding measured conditions (thus turning users into a mobile reporting service). Thus knowledge about and how to use these reporting functions will be shared and transferred between users visiting such a MIS. Assume users are moving in and out of WLAN coverage to an umbrella GPRS system at a rate of once per 200 seconds (assuming a coverage area of 200 meters at a walking speed of 1 m/s), with a population of 100 mobiles (users), then on average a user will move in and out of coverage every other second, and request or emit connectivity reports. The MIS may also offer a location prediction service, offering to process this information enabling mobiles to project an “always-best-connected” path through areas with widely varying data rates.

8.3.7 Service Prediction

MSK and the components of the ACM, in combination with XSP enable the simulation of application behavior and thus enable us to predict service behavior. Given the fact that we know the relevant piece of knowledge regarding a piece of equipment or service, we can build a simulation model with similar entities and generate events to predict the behavior of a larger population or to see how this application consumes network resources in certain situations. For instance, the aggregated models are synthesized and aggregated by a user and the entities that the user had encountered (in Section 9.5) can be copied to entities with ACM in a simulator where stub code replaces the APIs which otherwise would be needed for invocation of communication sessions. Hence, scenarios can be invoked by sending events into this model allowing an application developer to observe simulated behavior off-line.

8.3.8 Security

Knowledge must be packaged in a secure way and given to trusted parties [79]. It requires some thought regarding how to make this transparent and lightweight. This is similar to problems in

![Figure 8-6. Dissemination of MSK](image)
e-commerce that also need to provide an exchange of services where parties can be assured of the sender’s identity (non-repudiation), the service has not been tampered with and arrives unchanged, and that another party has not been able to read the content during transport. Solutions available that offer end-to-end security can utilize for instance IPsec or a PKI infrastructure. See also Subsection 7.5.3 on page 111. See also Subsection 9.4.5 on page 138 on the use of a firewall, NAT, and DHCP in order to raise the level of security for a local access provider while preserving SIP and XSP communication.

8.3.9 Privacy

In addition to data protection (Subsection 8.3.8), the issue of privacy protection in computer networks is rapidly becoming one of the most important issues on the Internet today. More and more Internet sites are collecting personal information from users and exploiting the ability of mobile communication service providers to determine the geographic location of their users, offering location information that is being used to provide location-aware services. A user needs to be able to preserve location privacy, which is the ability of a mobile node to conceal geographical information from third parties while the user is on the move [40]. The models for data and location protection are compatible with the service architecture since they allow end-to-end (peer-to-peer) communication.

8.3.10 Preserving Peer-to-Peer Networking (Firewalls, NAT)

The service architecture is based on an end-to-end model of communication, which supports the mobility of devices, services, and users. At least two entities that are commonly used on the Internet do not support the end-to-end model, in specific firewalls and Network Address Translation (NAT) [139]. Firewalls restrict the use of ports and it is not uncommon that only ports 80 and 8080 are open for HTTP connections via TCP. Similarly, NAT (invented to circumvent the problem with exhaustion of IPv4 addresses and used in for instance GPRS) sits between the public Internet and the network it serves, and works by rewriting IP addresses and port numbers in IP headers on the fly so the packets all appear to be coming from (or going to) the single public IP address of the NAT device instead of the actual source or destination. Conversely, NAT will not allow applications to send IP addresses or port numbers hidden inside their data packets, where NAT can’t properly rewrite them, and for security reasons, some NATs only allow incoming traffic from an outside address if an outgoing packet has already been sent to that outside address, which means that entities behind different NATs can't open up connections to each other.

For firewalls with or without NAT the solution is to design an application level gateway (ALG) [144]. Such modules are available for instance for Linux based firewalls with NAT capabilities (IP-masquerading) [141] enabling peer-to-peer networking between clients on either side of the NAT and also for multiplayer online games [141]. In summary, several solutions are available in order to deal with firewalls and NATs, and mainly proxies, application level gateways or design rules for protocols [137]. Within the context of this dissertation, I assume an application level gateway (ALG) for SIP [133,144] to be sufficient, when XSP is mediated via SIP extensions. Other considerations and design choices are the subject of future development efforts.

8.3.11 Trust and Control

The issue of trust and control is directly related to security issues. Security addresses mechanisms for disallowing certain interactions between parties or access to information which covered in the previous subsection. Trust and control is about the semantics of these mechanisms, which is how these security mechanisms are put to use for establishing trust relations between parties or establish relations such that one party is allowed or disallowed to control the other. The issue of trust and control has made it difficult for users to accept communication where interaction with objects in the environment became explicit. Clearly, these are non-issues in homes or offices, where users, devices, and resources generally have trust relations by default, as they feel secure when protected by a
firewall. This is not the case when for instance a user is visiting a building or even more importantly visiting another access network in a conglomeration of public mobile networks (for which I used the term $\pi_G$ in earlier chapters). For the user visiting the building, as well as the network in this building, as well as the visitor in an access network that this person is new to, the question is what could convince these parties to trust each other, and for the services, what is required for the ACMs to allow or disallow interaction (i.e., negotiation via XSP). Some preliminary conclusions are that establishment of a trust relation requires the secure endorsement from a third party, along similar lines as outlined in [66]. However, the ACM could very well have cached trust information so that we could establish trust relations without this third party to be available on-line, as long as the parties that wish to establish a trust relationship can determine that they refer to the same third party. The secure caching, exchange, dissemination, and exchange of information for the establishing relations for trust and control is vital for the acceptance, and this success, of pervasive computing applications and creating services for $\pi_G$ alike. This is an area that merits further study. Within the scope of this dissertation, it suffices to say that MSK which models entity-relationships, is well suited for extending these relationships with information concerning trust in combination with the support for access control that was mentioned in Subsection 7.5.3 on page 111.

8.3.12 Power Management

The model that implements the active context memory can also be used to program the power management (“I am out of the office, stop listening to me, and switch to standby, turn off the display. Wake up in case something very important comes up. I’ve come out of the meeting - wake up”). The mobile can learn the behavior pattern of its users and respond in an energy conscious way.

8.3.13 Computational Resources

The power management schemes can also benefit from being allowed to distribute the computation such that the mobile has less computation to do. For instance in the case of speech recognition the mobile might benefit from utilizing a remote server that does the recognition. Furthermore, we might also decide to program the personal agent and the active context memory in the mobile to have a standby mode. In such a standby mode the client is not entirely off-line, but rather it forces the master copies of the agents on the agent server to do the processing and only involve the agents in the mobile when specific user intervention is required (for instance an urgent call).

8.3.14 Persistence

Consequently, there can optionally be an agent server in any Mobile Interactive Space where master-copies of these mobile agents reside, for use when the user is logically off-line. Re-synchronization of mobile data must be done to ensure consistency (e.g., using Java object serialization or in the future perhaps SyncML [197]).

8.3.15 Scalability

The obstacles in scaling these applications are not an inherent property of the new infrastructure. Proper dimensioning of these networks can accommodate millions of users in large-scale systems. The cost of transmission is low and bandwidth is simply not the problem, even where VoIP and scalability are concerned, which is shown in my papers [72,74,85] concerning VoIP over wide-area radio access networks. Furthermore, the architecture allows us to scale the solution and physically distribute the functionality, for instance, in the case of a Contact Center [69,70,71], we showed this to be possible.

When we examined scalability at different layers of the architecture in Section 4.4 on page 55, we found that the service architecture scales very well, provided we take precautions in the Mobile Knowledge not to flood the network disseminating knowledge and sending event in promiscuous
modes. One one hand we can rely on users being prudent in their use of available bandwidth and batteries. It is however possible that entities display unintended (emergent) behavior, which may not directly visible to a user. Therefore, we may need to allow operators to enforce policies in the Mobile Knowledge Layer by demanding users to accept such packages prior to when the user is granted access (see Section 9.5 on page 142).

8.3.16 Step-wise Introduction

We found that the society of agents’ paradigm [58] that we adopted in the prototype has problems. We simply cannot assume that the entire world of communication adheres to a common agent language. Neither may we expect that there will be a similar universal accepted language for the exchange of context-aware knowledge to be used above the Active Context Memory for the very same reason. A much more probable approach to success is to assume a common basis for the publication and usage of services on a lower level, where intelligent objects or agents are confined to tasks like representing local higher level behavior. The communication protocols are then used to publicize the properties and resources of the higher level services involved for further processing by other software. Examples of middleware for sharing services over the Internet are CORBA [151] and JINI [195].

8.3.17 Network Economics

The design of service architecture and its components scales results from mobile computing and merges these so as to provide these adaptive services of mobile users in large scale mobile networks. The added cost of these components that are used in these solutions, is low as demonstrated in our prototypes that I describe in the next chapter. A word of warning is needed here. Caution must be taken so as avoiding including unnecessary optional components to either mobile devices or cheap access points as they relatively cost sensitive. In addition, mobile devices have a limited power budget which warrants further study on the relative power consumption needed for different smart media services versus the relative gain in battery life due to distributed knowledge of how to avoid unnecessary communication and computation in the mobile device (e.g., if the mobile device learns that the necessary bandwidth will not be available or required for another hour, it can agree with the network resources to stop on-going transmissions and suspend itself for an hour before making a new attempt. In addition, relative to the application we should monitor the performance of application assuring the success of the ACMs to restrain promiscuous eventing by filtering and scoping so as not to drown the network with unnecessary events, which will not of interest to others than a very limited group of entities.

The prototype results in the next chapter are very encouraging indeed, in confirming what has been stated in this subsection that the added cost of infrastructure in order to deliver adaptive and extensible multimedia services is very low, facilitating rapid deployment. Even more importantly, these experiments show that we can greatly relax the requirements for delivering the services.

8.3.18 Packaging Service Behavior

Service behavior can be packaged as a conscious effort of a designer (equipment manufacturer, application provider) or be done automatically as the ACM aggregates MSK pertained to specific interactions (e.g., opportunistic download of multimedia) in isolated clusters of MSK and identifies these as candidates for packages. In the later case, packages fulfill a purpose of introducing another level for users to store and retrieve their data, which will make this process more secure and easier to manage (as we can monitor the arrival of whole chunks of data and attach check-sums and signatures to these volumes of data).

More importantly, packaging of MSK, enables the equipment manufacturers, application providers, and network operators to include MSK with the equipment or offer add-ons to packages that the user already has, taking into account the considerations in Subsection 8.3.11 on Trust and Control. In
addition, these parties, need to take examine combinations of these packages for Feature Interaction, which can be done (with certain limitations) automatically using the support in the ACM — see further Subsection 8.3.2.

8.3.19 Service Repositories and Management

8.3.19.1 On Knowledge Maintenance

Previous attempts to create a global world of intelligent objects (e.g., an Agent Society [58]) led to interoperability questions necessitating the standardization of objects and methods in the universe of discourse (as is the case with the approach taken by FIPA). Similarly there exists a similar ontology maintenance problem for the Semantic Web. MSK in combination XSP and the learning & inferencing support in the ACM, takes a different approach that, although it will not be more successful in a highly static environment such as formalized business-to-business transactions, it will perform much better when creating a mobile environment with dynamic services, as the proposed architecture removes the requirement of having globally known and standardized ontologies. In addition the ACM will be able to support maintenance of ontologies and store these on MSK Servers, thus the models can successively be refined even on the scale of mobile networks, in parallel to the refinement of models as described in [111] for mobile devices.

8.3.19.2 Naming, Localization, Taxonomies

A property which is very useful for the retrieval and building of taxonomies of mobile service knowledge, in order to facilitate the manageable and maintenance of mobile service knowledge, are inclusions of URIs in the specification, enabling us to rely on a DNS infrastructure for network-wide naming and localization of MSK. Localization of packages MSK within a repository can then adhere this style, and the building of taxonomies and be mapped to a directories in a file system and made accessible via HTTP, for interoperability with other systems (which do not support XSP), in particular those which rely on the use of RDF.

8.3.19.3 Managing Diversity

The packages, which were discussed in Subsection 8.3.18 can be stored in and retrieved from mobile service knowledge repositories, using the mechanisms offered by the components of the service architecture that have been described in earlier chapters. As long as the usage of packages of Mobile Service Knowledge has a (semi-)local scope (i.e., not shared or only shared between a provider and a user), the task of managing this knowledge is restricted to providing support for packages in our foreground/background handling (Subsection 6.3.11 on page 101) and examining merged and combination of packages for feature interaction (Subsection 8.3.2), and resynchronizing records for changed data. Operators of mobile networks already manage our customer data and adding these mechanism will not be difficult or change their model of operation.

However, when we move to a deregulated heterogeneous wireless network, where users may choose between a multitude of (local) network access providers, Internet Service Providers, Application Providers, etc., we encounter new requirements, to be able to deal with similar or duplicate behavior, and avoid suboptimal performance, because more and more Mobile Service Knowledge is accumulated in the network which becomes more and more diverse and threaten to burden the mobile devices. The service architecture has relaxed the requirement for network-wide standardization, publication, and adoption of service knowledge for it to become operational (cf, FIPA, ‘.NET’, Semantic Web), but the advantages of a more flexible approach with a greater freedom of choice for the end-user which has been enabled by the service architecture also requires measures to compensate for the increasing volume and entropy which increased diversity threatens to bring about, e.g. having a multitude a packages with slightly dissimilar service behavior stored in the mobile devices wastes memory, and consequently also power, and has already wasted bandwidth for loading
these packages onto the device.

Therefore, we need support for detecting similarities and proposing streamlined packages, for storage on publicly available mobile service knowledge repositories. Thus, I envisage the need for MSK package profiling tools and automated librarians, as well as the need for MSK ferrets that report on MSK to these automated tools. These knowledge maintenance tools will even be useful for maintaining, optimizing, and profiling MSK that is shared within an ISP’s network (see Figure 3-4 on page 35).

8.3.20 Service Data-mining

Provided that users agree to emit logging information, a trusted applications provider can collect this in order to do data mining. In the simplest case, encapsulated knowledge pertaining to a specific context or entity can be extracted. Once collected from a population of users, this data can be mined for statistical patterns (e.g., what locations where most frequently visited, and regarding mobile group awareness what we the frequencies of interactions). In more advanced cases, knowledge containing reporting functions can be built in. For instance, regarding the case study in Section 9.5 on page 142, the credential provider can provide as an extra data-mining service access to collected access records and offer this as a new service in order for application providers to learn where their customers are and determine an efficient strategy for proving content in proxies for mobile users according to Section 9.7 on page 158.

8.3.21 Manageability

Since we base our applications solely on IP, we can easily adopt the existing techniques for managing distributed entities, i.e. SNMP. Installation, configuration, and management of the applications can be done entirely by end-users. Any other approach, such as packaging the network access with applications as in numerous field trials, is doomed to fail. Furthermore, aside from defining domains for global communication identities, here represented by a SIP-URL, no central customer administration is necessary. To a large extent user communication identities can be created in similar self-service dialogs, as is the case for e-mail addresses today. Intelligent communication behavior, implemented in Active Context Memories that are attached to these communication identities, may be created at the same time.

However, the previous section put forth reasons for operators requiring users to accept MSK packages that contain policies for service behavior in their network, in order to restrict or limit the damage of unintended rampant negotiations. In these packages, operators could (even though there is no necessity) demand to include MSK, which grants them rights to monitor the behavior of the ACMs and use of XSP, in order to properly dimension and configure network resources. This points at the need to further examine trust and control.

8.4 Chapter Summary and Conclusions

This chapter discussed the network aspects and requirements for how the open service architecture and its components may be applied, and analyzing the impact it has on the way existing operator provide services and opportunities for new operators. This was followed by a discussion regarding what requirements these new applications and services impose on the network (especially when wireless access occurs) and the implications for the communication, providing evidence that the open service architecture indeed constitutes a scalable means for providing services for building smart spaces on the scale of large mobile networks. Furthermore, this chapter investigated different aspects regarding trade-offs regarding different scenarios and distribution of functionality of the service architecture framework, which need to be taken into consideration in order for different categories of operators to be successful.

The conclusion is that the service architecture and its components scales, and enables existing and
new operators to deliver services in new ways. This occurs because we removed the limitations that prevented service growth in existing (mobile) telecommunication networks, to the extent that anyone (fixed or mobile) can offer a service, in an ad hoc fashion.
9 Enabling Mobile Communication

This chapter first examines what new classes of mobile communication have been enabled. For each class of mobile communication these sections show novel contributions of my approach. Second, results from prototyping efforts are presented demonstrating the success and feasibility of this approach for enabling mobile communication. Finally we discuss some high-level implications of this approach for future modes of user communication.

9.1 Introduction

Mobile Awareness through Mobile Service Knowledge (MSK) interpreted in the Active Context Memory (ACM) and negotiated or exchanged via eXtensible Service Protocol (XSP) is a powerful concept. It enables us to reason about the communication conditions that we encounter and respond accordingly. The ability to anticipate changes in our communication enables the device to hide a number of types of resource limitations (for instance pre-caching can hide subsequent low bandwidth conditions). Requests for printing when we are off-line can later be realized when we have located printing resources in the network; thus when we have suitable conditions, then we can carry out the print request. When we engage in multimedia sessions with a person, may be reminded (based on prior context memories) of other occasions that we talked to this person and of the topics that were discussed then. We can also adapt our behavior, such that in cases when the bandwidth of our connection is reduced, we can utilize more aggressive filtering and/or encoding in order to preserve the current session(s) while reducing the required bit-rate. In other cases we might completely change the mode of communication, e.g. from multimedia to text-based chat. Far more elaborate examples are possible, where users exchange knowledge in a communication situation such that one person shows the other person how to use resources modeled in a virtual space. In other cases combinations of resources that exist within range can be utilized via ad hoc communication. For instance multiple devices in a living room act in concert as an ad hoc multimedia answering machine that is triggered by our appearance.

Thus, the properties of MSK (which were examined in detail in Section 5.3 and Section 5.4, in combination with the support offered by the Active Context Memory (ACM) and the eXtensible Service Protocol (XSP), for the interpretation, negotiation, management, and exchange of MSK) have some high-level implications on our system design, as well on the enabled scenarios. End points and intermediate nodes in the network can be extended with repositories for mobile service knowledge and reasoning, thus representing high-level awareness and decisions about what these components are trying to achieve. Given the service architecture and its components, the following capabilities have been enabled:

1. The system can adapt the mode of operation of the components or allocation of resources in the network in order to deliver the expected service to the user — this is described and exemplified in Section 9.8.
2. the system can make decisions about delivery of relevant content relative to the context of the user or the device — this is described and exemplified in sections 9.6 and 9.9 — or
3. make decisions about aggregating resources in the user communication, taking into account the user’s context or the goal of the application — see Section 9.5

Subsection 9.6.3 contains further ideas of applications, which could extend our prototype work.

9.2 Modes of Human Computer Interaction

9.2.1 Existing Modes

The service architecture enables the use of a richer and more flexible interface as it enables the point of service of service integration to be moved out of the network to higher layers in the end-points. Hence it enables the components and end-points to extend the mode of user communication with media components and awareness to what the user is trying to do. In Section 9.6 I exemplify how we can extend existing modes of user communication by making taking into account the context of the user and the goal of the communication in order to make communication decisions.

9.2.2 New Modes

By virtue of the relaxation of the prior limitations the service architecture enables radically new modes of communication and human-computer interaction. They can be classified as follows - although the list must be regarded as open ended [128,146]:

Augmentation Modes of human computer interaction can now be modeled and extended in any suitable fashion by presenting overlapping views of reality (i.e., augmented reality), usually employing wearable devices, having a see-through heads-up display and wireless access. The user interface can be dynamically enhanced with new features, where voice is just one of the basic services. Therefore the range of user action can be extended in various ways, where the term augmented-reality describes how the communication emphasizes or complements observed real objects or events. The ACMs can also infer what might happen providing what-if cues to the user, which could be termed hypothetical-reality. Furthermore, due to the ACM storing observed events and context we can present replays of past scenes. This augmentation of communication is described further and exemplified in Section 9.9.

Androids When the degree of augmentation passes a certain point wherein the user is integrated into the feedback loop of virtual or physical computational objects and vice-versa; this brings about a shift in the mental state of the user in which the user identifies themselves with a certain synthetic personality in the communication space that is larger in scope than the previous personality. This synthetic personality can either be totally digital or an aggregate of both digital and physical parts. Examples of the first are Avatars or maneuvering a physical robot remotely while logically teleporting oneself to the robot. Examples of the second are body implants where actions of the owner are visible as actions of the synthetic digital personae.

Affect Emotional Contexts. Grouping of MSK in meta-contexts corresponding to emotions. Affective Behavior

Sections 9.5 through 9.9 show examples of the above new modes, which we illustrate with our prototype results. However, the issues regarding Human Computer Interaction (HCI) design lie outside the scope of this dissertation. This chapter demonstrates that the service architecture is totally open to these new modes of man-machine interaction while preserving the old modes. It does so by analyzing a series of scenarios that instantiate the abstract service architecture that was described in earlier chapters.
9.3 Service Delivery

In this section I present an overview of different modes of operation delivering existing and new services. The overview exemplifies how the service architecture enables the delivery of new services or optimizes the use of communication resources for delivering existing services with pointers to later sections in this chapter where this is further demonstrated with empirical evidence.

9.3.1 Delivering Existing Services

9.3.1.1 Smart Media

Intelligent routing and Smart Delivery of Multimedia - see sections Section 9.7 and Section 9.8

9.3.1.2 Multi-device

Multi-device Applications. Multi-device scenarios are the result of the MSK becoming aware of the availability of resources, and sending events to these resources due to activation of their mobilets causing these to act as event adapters to the actual resources. For instance, a person with a call waiting entering a meeting room equipped with large display may borrow the public display (i.e., direct their video via the local broadband network to this display), while audio and voice control is routed via their PDA, which is equipped with a microphone and audio output.

9.3.2 Delivering New Services

9.3.2.1 Swarms

When entities are organized into clusters where the action of one entity is propagated to the others not only as an event but also to accomplish a common task (by being input to a task in the ACM) in turn causing new responses by the same group of entities, we may speak of swarm behavior. Such eventing cannot cause all entities to exchange events since this would drown the network; however it may suffice if every entity communicates with its neighbors. This way we can easily envisage the dissemination and subsequent assembly of a urban bandwidth topology of mobile groups — without requiring a central server, this might be used for subsequent exploitation to realize smart delivery of multimedia (Section 9.8).

Swarm Behavior for Mobile Devices: As MSK can be distributed over a multitude of hosts (and moved between them when necessary in whole or partially), we can also further aggregate MSK in a distributed fashion, such that this multitude of hosts can achieve a common goal. For instance, when all mobile devices aggregate information about the communication conditions regarding data rates, latency, etc. over time in different locations, then they may arrive at collective decisions on optimal routes for the exchange of multimedia information between them, relative to the requirements of the multimedia.

9.3.2.2 Teleportation

The events (stimuli) that the ACM and therefore the aggregated MSK in it receives, may cause the MSK to request itself to be moved to another host by means of the eXtensible Service Protocol (XSP). When a cluster of mobilets and meta-contexts including their state of execution, are moved to another host, we may speak of teleportation. For instance, when we move MSK for our home to a hotel server, then events from devices similar to the ones we have at home may result in similar behavior. Thus, our home environment may be said to have teleported to us. In addition, when for instance a person moves from his/her home to a hotel room with XSP enabled devices, then the mobilets that correspond to home devices including their accumulated knowledge can and will be mapped onto the devices that are present in the hotel room.
More exciting scenarios are possible where we could send a physical networked object with sensors to a customer (for instance a toy to children). Interaction with the toy will reveal to the toy the communication identity of the owner. As a member of the household the child has access to and knowledge of how to use certain multimedia equipment. This knowledge can be transferred to the toy’s communication context so that it now knows how to use it. Thus the toy has been able to, in an ad hoc way, extend its range of communication and may now start to interact with all kinds of appliances in the house, provided they are present in the model, and that they trust this new entity. Therefore a key element of security is to introduce this toy to the family’s trust manager.

9.3.2.3 Mobile Robots

As the ACMs are able to store and record all events and contexts perceived by the user, it can remember behavioral patterns and it has in principal unlimited access to our input and output. Hence, it can replay our actions and also collect and remember responses to these actions, including how we would have reacted in a what-if analysis. Furthermore, the ACM can move or export its knowledge and execution via XSP. Hence, we have enabled what may be referred to as a mobile robot, meaning an independently acting and goal-directed self-contained cluster of MSK that uses resources in the infrastructure for its own goals and that can carry out tasks utilizing resources/devices & move about in the network. Besides developing behavioral intelligence it may develop a cognitive model with certain explanatory powers. It is outside the scope of the dissertation to show whether artificial intelligence is possible. However, I argue that the service architecture and its components enable the self-sustaining (see also the learning and repository sections 6.2 and 6.5).

This claim is sustained by the fact that today robots in real-time computer games utilize artificial intelligence techniques such as finite state machines and genetic algorithms for path-finding, steering behaviors, and a decision-making part resulting in the creation of entities that are able to co-operate in order to accomplish tasks, and display flock behavior. Therefore, the entities in the type of applications that have been studied in this dissertation and prototyped in the sections of the remainder of this chapter will (due to the subsequent incorporation of these machine learning capabilities in the ACM) also be able to display similar behavior in mobile applications, such as co-operative behavior in order to deliver the smart media packets as described in [81] and Section 9.8.

In addition to mobile robots I envisage the creation of mobile brains, see further Section 11.4.2 on page 182 in Chapter 11 on Future Work.

The remaining sections of this chapter describe results and experiences from prototypes and a wireless testbed which provide further corroboration for what has been enabled by the open service architecture and its components for providing adaptive mobile communication.

9.4 Experimental Network and Results

We have built an experimental pi-generation (πG) wireless testbed by extending Internet42, an existing Gigabit-Ethernet IP-network [82,83,84] (Figure 9-1). The project involved several parties: Ericsson Radio, Royal Institute of Technology (KTH), Telia, and Brf. Bågen. Besides points of presence at research facilities (Ericsson Radio, KTH, and Telia) in the Stockholm suburbs of Kista, Älvsjö, and Farsta; Internet42 also has a point of presence in the center of Stockholm where it provides, at low cost, 100 Mbps network Internet access to each apartment in a large housing co-operative, Brf. Bågen [165]. We have extended the services of Internet42 [177], by adding 11 Mbps wireless packet data access points (IEEE 802.11b), agent servers, media servers and content management, and voice gateways (VGW), thus we can provide anonymous Direct Internet Access (DIA) [66], support for device mobility (Mobile-IP), and service mobility (SIP). We have recently added GPRS to our test bed. The functional components are further explained in the sections below.
In April 1998 the housing cooperative Brf. Bågen in central Stockholm installed a local area network in all 261 apartments and in all companies located in the buildings [177]. The main use of the LAN was initially to provide Internet access through a leased 2Mb/s line.

Today, Gigabit Ethernet is used both to connect the buildings to the Internet backbone and between the five Ethernet switches, which provide each user with a 100 Mb/s Ethernet connection to the Ethernet switch. The housing cooperative acts as an operator with the following distinguishing characteristics: (1) Users get real IP-numbers, either statically (for servers) or dynamically through DHCP, (2) there is no firewall to the Internet, and (3) there is no restriction on traffic, neither between the users nor to the Internet.

The only local services provided are mail, local personal web pages, and local news. Currently 56% of the apartment owners are actively connected to the net. A few companies are also connected, and share the outgoing bandwidth with the apartment owners. The residential LAN infrastructure has worked very well with the exception of a few prolonged interruptions of the Internet connection, (which led Bågen to change ISP after an open tender). It is important to note that this was possible and relatively easy to implement due to the fact that Bågen owns the LAN.

### 9.4.2 Wireless Access

Extending a fixed ethernet network with (IEEE 802.11b) wireless-LAN access points near the points of presence in research facilities was straightforward. Adding wireless LAN to a housing co-operative and thereby providing wireless access to Internet over its infrastructure in a public space was a different matter as it confronted the housing co-operative with both technical issues (mainly
security) [113] and non-technical questions (concern about antenna aesthetics). The effort and cost to provide broadband wireless packet data was very low. As usage grows we can add access points. Initially, with a single access point we obtained good coverage in a large public space at low cost (Figure 9-1), as the cost of hardware was $2600, and the area covered was 200m in radius = 126000 m², the cost was $0.02/m². Users share up to 11 Mbps of bandwidth via a single access point, but wireless LAN technology (802.11b) easily allows us to add access points as the user density and demands increase. Monitoring throughput during video conferences (between a Sony Vaio C1 Picturebook equipped with WaveLAN cards (IEEE 802.11b capable of transmitting at 11 Mbps) at Bågen and the measurement carried out using an identical piece of equipment at Ericsson Research in Kista. We observed 80-90% network utilization over the wireless link when sending non-compressed video of 16 bit color images 320 by 200 at 10 frames/second. This data traffic with overhead (IP-headers) is roughly to 10 Mbps, thus this load is what one would expect. The user experienced no problem, even in this worst case scenario, and audio was easily accommodated using the remainder of the bandwidth. Clearly, with more prudent and intelligent strategies than this brute force approach, communication for many users and simultaneous users can be accommodated. Section 9.7 describes another experiment with a multimedia prototype within which we only needed 1.2 kbps to transmit voice over the wireless link — see Section 2.3.1 and [72,74,85].

9.4.3 Direct Internet Access

Wired Equivalent Privacy (WEP) in IEEE 802.11b has a dual purpose of authenticating users and providing data encryption with the following disadvantages: (1) WEP differs between manufacturers, (2) WEP encryption keys must be manually distributed, (3) WEP is set on a per-network basis rather than on a per-user basis, and (4) Windows-based machines must be rebooted after a key change. Alternatively, the WLAN infrastructure can be complemented with an authentication mechanism based on pre-shared or certificate-based keys. However, this approach precludes anonymous roaming access. Therefore we used a third method Direct Internet Access (DIA) [66], which provides anonymous authentication and allows the access provider to charge via eCash. This approach thus makes access authentication keys redundant, and allows simple roaming access. Consequently, there is no reason the WLAN access points to do accounting or administration of users. Additionally, end-to-end security is ensured using IPsec and IKE [113].

9.4.4 SIP naming, localization, and session invocation

A SIP redirect server allows end-uses to register with a SIP URL and enables others to send them invitations to multimedia communication (enabling personal service mobility). Thus, assigning these identities to Personal Agents allowed us to leverage its functionality to easily implement (1) remote customer control of personal messaging (via web pages) and (2) Internet Telephony (e.g. diverting calls when in a meeting), the voice gateway allows us to locate agents locally or remotely as SIP URLs by identifying telephone numbers and vice versa. When we allow the personal agent to monitor incoming calls to its number via the telephony-GW using a group number, then a consistent and complete (i.e., messaging, Internet-, and switched- telephony) solution was achieved for personal communication.

9.4.5 Security

SIP invitations can be sent through firewalls and despite NAT [144], which might allow a local network access operator (e.g. Brf. Bågen) to increase its security while preserving all services. A firewall in combination with NAT and DHCP makes it considerably harder to find the addresses of internal machines and open ports in an attack. However, SIP-based multimedia communication and negotiations between entities by means of XSP based on SIP-extensions are preserved.
9.4.6 Mobility

Strategies using Mobile-IP or other network mobility protocols can be used to support handoffs [24,153]. We used the Mosquito Net Mobile-IP stack [161] to enable our devices to do handoffs between GSM-data and WLAN, in order to investigate the feasibility to do VoIP handoffs. These attempts proved to be unsuccessful due to various reasons: GSM-data session setup times through a dial-in connection were too time consuming; this is fixed by using GPRS. However, we found that infrequent agent advertisements (minimally 1 sec. delays - RFC 2002) overshadow the 500 msec latency in the GPRS air interface. Thus a modified approach is needed (e.g., with micro-mobility). A separation is needed between mobility for voice and the mobile device, so as to circumvent unnecessary delays due to triangular routing, where location changes are used to send voice packets to the new address, either via SIP or simply using RTP, thus resulting in minimal delays. See further the earlier discussions of this topic in Section 4.9.

9.4.7 Handoff Measurement

Figure 9-3 and Figure 9-4 show early results from measuring delays during handoff between two IEEE 802.11b interfaces of a single access point (AP) with 42.4 Kb/sec UDP traffic (50 pkts/sec., 64 bytes payload). The handoff was forced by weakening the signal of interface 1 (Figure 9-2). No packets were lost, the handoff incurred an additional peak delay of 157 msec, with this delay decreasing linearly over 9 packets, and leveling off to the previous value, since the access point's buffer quickly emptied after the handoff. With the current setup of our measurements (Figure 9-2 showing the AP with two wireless interfaces) it could not decisively be determined what was the dominating contributor to the extra delay of 157 msec.

However, the numbers can be explained by assuming that the sum of the delays before packets could be delivered was ~180 ms and the inter-arrival time of these 9 packets was near zero (i.e., they are delivered as a burst of packets from the second wireless interface to which the mobile moved once the communication has been re-established — see Figure 9-4):

1. The mobile and the AP communicate roughly every 100 ms [31,163] so the mobiles know they can still hear the AP and vice-versa, and if the mobile can't hear the AP (and the reverse) it will not know for 100 ms (since it has to miss at least one such event).
It will therefore take on average 50 ms until the mobile hears from the other interface (and on average 1/2 of the beaoning interval before it hears a beacon).

This explains a delay of 100-200ms (with an average value of 150ms) for the mobile to move. The arrival of a new packet 0-20 ms (an average of 10 ms) causes the AP to want to transmit something to this mobile and therefore the average time for communication to restart time for reconnection is ~160ms.

This explanation is especially plausible since the handover was between two wireless interfaces in the same access point that share the buffer. At the receiver the packets subsequent to the handoff arrive with a delay of ~180 msec minus N times 20 msec, where N is the index of the packet after the handover, since the 9 packets were transmitted at 20 msec intervals. Handovers between wireless interfaces in different APs will add a delay if the first access point is to transfer its buffer's contents to the second AP, but this added delay will not change these results significantly, since this operation could occur at the speed of the wired LAN connection. However, the spanning tree calculations could take a considerable period of time if the bridged network has many bridges and ports. For a description of spanning tree calculations and suggestions for improvement of their computation times see [130]; Further measurements and experiments are needed in order to gain full insight in the conditions for hand-overs in WLAN networks - see Section 11.2, [40,157].

Irrespective of this, the extra delay of 157 msec during less than ~180 msec (9 packets) is something that could be addressed by the codecs in the end-points. Alternatively other techniques for rapidly detecting changes in network connectivity and application layer responses to changes in connectivity could allow such handovers to be unnoticed by a user of either VoIP services (see figure 3 on page 5 in [35]) or streaming media services. For example, in the case of one-way streaming media a jitter buffer of just 200 msec would have hidden the above handoff. Perhaps in the future an application will be able to notify the local mobility manager when it fails to receive an expected packet, in this case this could trigger the mobile's interface to search for a new AP after only 20-40 ms (depending on how aggressive it wished to be).

9.4.8 Quality of Service

As there was ample bandwidth, the use of speech codecs was unnecessary from a QoS perspective. There was no perceivable packet-loss from the perspective of the end-user. However, in areas where the signal is weak, we may benefit from using robust header compression [17].

When end-to-end security needs to be guaranteed, then IPsec is an obvious choice. When speech and signaling use different ports then the increased header length can be dealt with separately by applying ROHC [17,37] to this stream of IP-packets without requiring a trust relation to be established between the mobile device and the access point. This allows us to utilize the direct Internet access strategy without requiring AAA-functionality, unless it is necessary for commercial or operational reasons. See further the earlier discussions of this topic in Section 2.3 and Section 9.8.

9.4.9 Capacity

In addition, capacity and spectrum efficiency will benefit from using robust header compression [17], as IP-headers incur a considerable overhead with respect to the size of content for streaming
audio. Furthermore, speech compression will further up the number of simultaneous voice users. In ideal cases, with robust header compression, and compressing 16 Kbps PCM to around 6 Kbps G.723, the number of possible simultaneous voice users, in a single 11 Mbps cell could be well over one thousand, corresponding to 105 users in a macro cell, with the possibility of adding more access points if more capacity is needed. See further Section 9.7 and the earlier discussion of this topic in Section 2.3.

9.4.10 Hosting and Interworking

An important aspect for parties such as Brf. Bågen whose main focus it is to make connectivity available in their infrastructure, but whom are not interested in directly operating services, is that they should be able to outsource hosting of substantial parts of the functionality. This is supported by our architecture, as it is of no concern where most of the components are located as long as they are available on the Internet. All this functionality could even be packaged as a do-it-yourself PiG kit, since management of the functionality is not necessarily more complex than maintaining a web site.

Interworking with other Internet access providers, both wireless (e.g., Telia Homerun [201]) and fixed networks needs to be addressed at two levels. First, the user must be allowed access to and to be able to roam between networks. There are technical solutions available for both. Second, there must be an agreement between different parties, which describes who decides to allow roaming between various networks. This can be supported by a clearinghouse, such as Excilan proposes to be, thereby relieving parties of managing mutual agreements.

9.4.11 Summary and conclusions based on the experimental network

This section described real world experiences from extending an existing Gigabit Ethernet infrastructure with wireless access (mixed WLAN, GPRS), such that it could independently negotiate Internet connectivity (from different ISPs) and provide access and services to users [55]. We could provide voice services in this infrastructure with very little effort, and this infrastructure offered no limitations regarding extending it with the components of the service architecture as described in previous chapters in order to provide new services (which will be described in sections in the remainder of this chapter). The recipe for replicating this (in order to enable others to provide these new services to mobile users) is very simple:

1. Package functionality (e.g., voice gateway, SIP server, agent server, etc.)
2. Negotiate fixed Internet access (e.g., xDSL, cable, fiber, etc.)
3. Decide whether you want the functionality hosted.
4. Just connect the antennas and put them on your roof.
5. Negotiate your service level with the clearinghouse and have it announced.

We provided wireless LAN (IEEE 802.11b at 11 Mbps) based public access to the Internet in various public locations in Stockholm, one of which was the housing co-operative Brf. Bågen. Thus, we have shown that providing functionality to provide end-users with broadband end-to-end IP communication over wireless links using direct access to Internet is feasible, cost-effective, and enables adaptive mobile multimedia communication. This functionality can be packaged or outsourced (the later enables use by end-users of organizations whose focus is not primarily to own or operate networks). Thereby, we created a public broadband wireless Internet infrastructure that is not regulated and can in principle be used by anyone. This unregulated infrastructure provides connectivity to services that, irrespective of whether they are located locally or elsewhere, are not part of the network. Any transactions between the end-user and the application service provider are conducted without the network access provider having any knowledge or role. On the other hand, an application service provider may benefit from being able to get support from the operator for end-users in order to maximize performance of their service in these hotspots. Such a scenario has

been outlined in [66], in which case it is plausible that the provider of such a hot spot will be compensated, and thereby creating additional incentive for putting up such networks.

In conclusion, by virtue of our experimental results, we claim that development of infrastructure for 3G should be aligned according to the criteria that were discussed in this dissertation. Furthermore, R&D efforts regarding applications should focus on enabling mobile multimedia communication in such a deregulated infrastructure by adopting a service architecture that promotes deregulation on an application level [80].

In the following sections of this chapter I will describe experimental results from applying the service architecture for creating and providing mobile applications, which enable users and mobile devices to negotiate Internet access and services that take into account user context and communication conditions.

9.5 Ad Hoc Internet Access and Services

Here we will present results from a case study of scenarios for ad hoc (wireless) access to the Internet and services on the Internet in our testbed [82,83,84] (which was described above in Section 9.4 on page 136).

Three scenarios (two of which shown in Figure 9-5) demonstrate the use of the open service architecture and in particular ACM, MSK, and XSP to provide wireless access to the Internet and services to mobile users without requiring mobile users to have any advance knowledge of the topology or capabilities of the entities (which provide access or services) or requiring access and service providers to have advance knowledge of what services mobile users might bring with them.

Prior to the description of the scenarios in subsections 9.5.2 through 9.5.4, Subsection 9.5.1 below explains the discovery and synthesis of MSK that takes places during the negotiations between entities via XSP in these scenarios.

9.5.1 Discovery and Synthesis of MSK

In this model each node is an entity and for the purpose of this argument only the users, the WLAN access point, and the credential provider have been emphasized. The mobilets have the following templates:

![Figure 9-5. Testbed Model Overview](image-url)
/* <http://psi.verkstad.net/msk/wlan/tom> */
rogueWLAN @ "<http://psi.verkstad.net/msk/wlan/rogue>".
mobilet(tomAP) :: {
    type(rogueWLAN)
}

Figure 9-6. MSK Synthesizing Example

Figure 9-7. Negotiating Operational Models
Figure 9-8. Extension of Operational Models

/* <http://psi.verkstad.net/msk/user/theo> */
user @ "<http://psi.verkstad.net /msk/user>".
mobilet(theo) :: {
    type(user) &
    task(inet_xs)
}.

/* <http://psi.verkstad.net/msk/ user/dan> */
user @ "<http://psi.verkstad.net /msk/user>".
mobilet(dan) :: {
    type(user)
}.

Figure 9-9. Initial and End States
Other mobilet objects act as types (synthesized through negotiated communication of perhaps created by design):

```cpp
/* <http://psi.verkstad.net/msk/user> */
agent @ "<http://psi.verkstad.net /msk/agent>".
mobilet(user) :: {
  type(agent)
}.

/* <http://psi.verkstad.net/msk/wlan/rogue> */
user @ "<http://psi.verkstad.net /msk/user>".
cp @ "<http://psi.verkstad.net /msk/cp>".
mis @ "<http://psi.verkstad.net /msk/mis>".
mobilet(rogueWLAN) :: {
  type(mis) &
  inet_xs -> user &
  certify -> cp & /* relation operator */
}.
/* <http://psi.verkstad.net/msk/cp> */
user @ "<http://psi.verkstad.net /msk/user>".
agent @ "<http://psi.verkstad.net /msk/agent>".
ccp @ "<http://psi.verkstad.net /msk/cp/charas>".
mis @ "<http://psi.verkstad.net /msk/mis>".
mobilet(cp) :: {
  type(agent) &
  certify -> user &
  certifies -> rogueWLAN
}.
```

**Figure 9-10. Flow Diagram Scenario 1**
As a first step (after the discovery via XSP) towards the synthesizing of the mobilets, templates are extended with their inherited properties respectively. In the second step, the templates are extended with the event triggered methods from the relational objects corresponding to arrows in Figure 9-6 on page 143. Figure 9-6 shows these two steps for mobilets tom\textsubscript{AP} and theo (from the highlighted part of Figure 9-5 on page 142) in combination with their mutual relation inet\_xs.

Subsequent discovery, aggregation of profiles, and execution of relational objects enable the entities tom and theo to independently generate fully operational models in their respective ACMs mirroring each other (Figure 9-7 on page 143). Each model contains relations (generated from the templates within the local context) between the observed entities. As theo learns that tom\textsubscript{AP} relies on a third entity charas\textsubscript{CP} to verify the credentials prior to granting access, also theo and charas\textsubscript{CP} negotiate and extend their operation models (Figure 9-8 on page 144).

The initial and end states of this process are summarized in (Figure 9-9 on page 144) in order to emphasize the important point that XSP in combination with the ACM enables us to independently negotiate fully operational complementary models of MSK in the respective entities, allowing the entities to communicate, without requiring advance knowledge of the details of the intended

Figure 9-11. Flow of execution of MSK in Scenario 1
9.5.2 Scenario 1

This section describes a scenario (scenario 1 in Figure 9-5 on page 142) in which the discovery, aggregation of profiles, and execution of relational objects enable a user theo to negotiate and acquire access to the Internet via a privately owned WLAN Access Point tomAP and credential provider charasCP.

Once provided with access to the Internet theo discovers new entities (a Mobile Interactive Space tomAP and a user dan), aggregates information from their profiles to extend the models, with which theo connects to the services in the Mobile Interactive Space of tomAP.

Figure 9-10 on page 145 shows a detailed flow diagram of the signaling that takes place between the entities, where the general steps taken are as follows:

Prior to the arrival of theo, the other user dan registers with the mobile interactive space at tomAP. When theo arrives, he obtains a temporary IP address. During discovery theo finds another entity tomAP leading to mutual registration and exchange of models as illustrated in Figure 9-7 on page 143 (see also the source code and explanation of the synthesis of MSK during the discovery phase described in Subsection 9.5.1 above).

As theo learns of a third entity charasCP, these two extend their models with a relation certify, as tomAP (using the relation certifies) redirected theo to negotiate a certificate from charasCP. If charasCP denies theo a valid certificate tomAP closes the link, otherwise it assigns a new IP address which is valid during the remainder of the communication. Figure 9-11 on page 146 shows the flow of execution. Once being granted access, theo is able to discover a Mobile Interactive Space at tomAP (another of tomAP’s inherited properties) where theo discovers another user dan, whom theo decides to invite to a multimedia communication session via SIP.

From a security point of view, there is no reason for why Theo should be able to see Dan, or even if he could, view Dan’s profile. Clearly, mechanisms such as described in [142] are needed to establish the trusted groups, which have been assumed to coincide in this MIS.

Figure 9-12. Partial Flow Diagram Scenario 2
This example shows how the approach enables different parties to auto-configure spontaneous communication without requiring advance knowledge of each other’s existence or capabilities, thus enabling users to cope with heterogeneous wireless networks and access (e.g., 3G, WLAN, etc.).

9.5.3 Scenario 2

Equally importantly, these aggregated models are shown to be extensible in the following example when suddenly a “wireless hop” offers intermediate access in the same model - see Scenario 2 in Figure 9-5 on page 142 and Figure 9-12 on page 147.

Figure 9-13 shows the flow of execution in the extended MSK of the entities. Thus the innocent digital hitchhiker theo discovers two wireless hops and ultimately selects the one nearest tomAP through the new rem_xs relation that he has extended his local model (DAG) with. The tomAP access point has extended its inet_xs capabilities with acquired rem_xs capabilities, resulting in theo being
granted access, which theo discovers and connect to the tomAP mobile interactive space (tomAP acts by inheritance) and discovers available communication entities.

### 9.5.4 Scenario 3

In scenario 3, users theo and dan respectively discover each other’s presence in the Mobile Interactive Space. In the ensuing negotiation theo and dan discover that one of them has a CD which they decide to exchange — see Figure 9-14.

![Figure 9-14. Scenario 3: Mobile Interactive Space](image-url)

**Figure 9-14. Scenario 3: Mobile Interactive Space**

![Figure 9-15. Flow Diagram of Scenario 3](image-url)

**Figure 9-15. Flow Diagram of Scenario 3**

---

*Digital hitch-hiker* | *Digital hitch-hiker* | *Mobile Interactive Space* | *CD item in the MIS*
---|---|---|---
*dan* | *theo* | *tomAP* | *aCD*

- **layer 2**
  - newIPaddress
  - discover(theoId)
  - reg(misDAG)
  - pres/regMIS
  - pres/listMIS
- **layer 3**
  - xsp
  - xsp
  - xsp

*Already in Bågen*

*Access->New discovery*

*Discovery via MIS*

*And they SIP:ed happily ever after...*
Figure 9-16. Negotiation of MSK in Scenario 3

Figure 9-15 on page 149 shows the partial flow diagram and negotiation of MSK between the different entities resulting in the exchange of a multimedia object (content from a CD). Figure 9-12 on page 147 shows the relevant sections (omitting what is left unchanged) of the negotiations between the parties involved. As the “wireless hops” (e.g., users with wireless devices and battery capacity to spare) appear, upon discovery, the Direct-Acyclic Graphs (DAGs) are updated and tomAP, learning from the fountain that a new access method is possible updates its inet-xs relation. Figure 9-16 shows the extension of MSK in the ACMs.

Thus the innocent digital hitchhiker theo discovers two wireless hops and ultimately selects the one nearest tomAP through the new rem-xs relation that he has extended his local model (DAG) with. The tomAP access point has extended its inet-xs capabilities with acquired rem-xs capabilities, resulting in theo being granted access, which theo discovers and connects to the tomAP mobile interactive space (tomAP acts by inheritance) and discovers available communication entities.

9.5.5 Context Stacks

In paragraph 6.3.4.2 on page 95, the use of context stacks was presented as a means for tracing the
Chapter 9 Enabling Mobile Communication

9.5.6 Other scenarios

Other scenarios have been investigated such as, or in redesigns of earlier prototypes of a context-aware virtual office, location servers acquiring awareness of users and available public displays being closest to the user can display information relevant to the context of the user. More importantly, we showed how virtual entities could be created such as “meetings” (extensions of a Mobile Interactive Space). Users encountering such a “meeting” would negotiate communication and could participate without requiring advance knowledge of the constituent components - see Figure 9-18. Obviously, those users who lacked important capabilities (e.g., no audio) would be denied equal participation.

9.5.7 Section Summary and Conclusions

This section examined and described in detail how different entities used the eXtensible Service Protocol (XSP) to discover other entities, register and request profiles, and extend their Mobile Service Knowledge (MSK), thus accumulating fully operational models (corresponding to object-oriented event-driven semantic graphs of observed objects). Interpretation of these models in their respective Active Context Memories caused the negotiation of services. First, these components enabled the user to negotiate Internet access via an independent local access provider, who in turn introduced a third party to verify the user’s credentials. Second, the introduction of a wireless “hop” offered an alternative means to the user for connecting remotely to the local access provider, and caused the parties involved to spontaneously discover, register, exchange and extend their MSK respectively, thereby achieving the user to connect to the Internet via the wireless “hop”. Finally, granted this connectivity, the user discovered both a Mobile Interactive Space, a user in it, and negotiated services with this user (the sharing of a multimedia object).

These examples, distilled from the prototypes that were built and described in this chapter below, show that XSP, MSK, and the reasoning in the ACMs enable the entities to negotiate services:

- end-to-end, not requiring the intervention of a network entity (i.e., unbundling of the access method and services).

Figure 9-18. Entering a Meeting
• lacking a-priori knowledge about each other’s existence and capabilities (thus enabling the deployment of in successive steps, avoiding requirements of network-wide adoption for the solution to work).
• extending their behavior in response to changes in context.
• without a single operator being necessary to define and coordinate the service behavior.

In conclusion: these examples confirm that the approach meets the objectives stated earlier (see Section 1.4 on page 10) and enables the creation of user-centric computing & communication in extensible and adaptive environments.

In the following sections 9.6 through 9.9 below of the remainder of this chapter I describe two other prototypes that were built in our test bed (see section 10.4 above) which illustrate the importance of the open service architecture for enabling the new communication:

In [76,81] we demonstrated how mobile users were able to use opportunistic download strategies by negotiating the adaptive delivery of MP3 files through negotiation with proxies, who in turn had learned the topologies (bandwidth, location and proximity, utilization) of the available access networks. The content proxies co-located with access points) were able make intelligent decisions about their operation, preload content based upon observed user and network statistics and also to optimize link utilization in access networks by balancing bandwidth consumption versus user requirements.

Finally, we built a prototype [7,82,83,84] integrating events (sensor fusion) related to user movement (in the network by scanning access points, changes from Mobile-IP, and Cooltown sensors [90]) enabling the user to acquire knowledge about and the ability to use objects which both had a physical and virtual representation.

9.6 Virtual Office Prototype

In this section I describe how the hypothesis put forward in Subsection 1.3.4 has been tested by building a prototype of a VoIP-enabled Contact Center [69,70,71]. The prototype was built on our IP-based mobile computing infrastructure with wireless access (both wireless LAN and short-range radio links). This prototype enabled staff members to use the prototype application via the

Figure 9-19. Virtual Office Prototype
Chapter 9 Enabling Mobile Communication

infrastructure that is described below.

The basic communication services that we have focused on are messaging and voice services. An overview of our prototype is shown in Figure 9-19. The role and meaning of the different components are explained in Subsection 9.6.2. Subsection 9.6.3 explores the scenarios that we have implemented and discusses further extensions.

9.6.1 Application

The application for this system has been to improve communications both within an office environment and with others who wish to contact us (in various ways). Staff members were equipped with multimedia PC’s and an active badge. The badges made it easy to locate them and decide whether they were in or out of the office, just as was the case with the Active Badge system at ORL [152]. However, by routing communication automatically to the user’s VoIP terminal (running on either stationary desktops or wearable computers connected via a wireless LAN), staff members located outside of the office but connected via remote IP-access were perceived as being ‘present’ just like those within the office building. One of our project members was even located in another city. This sense of presence is important to the maintenance of the group, despite their physical distribution.

The staff could be contacted in primarily two ways:

- Using traditional telephony by dialing in to the VoIP enabled infrastructure, or
- Using the web with a multimedia enabled mobile computing system.

The use of traditional telephony gave no extra services, but neither did it remove any of the old ones. On the other hand, the use of the Web and new multimedia services enabled employees and customers to carry on a dialog, while simultaneously exchanging data objects. In addition, customers who had browsed the site and expressed interest in certain results could be routed to an employee with a competence profile that matched the destination of interest as expressed by the customer. The employee and their competence profile were in our case identified by the active badges. The dialog between the employee and the customer consisted of sending web pages with active objects back and forth in addition to full-duplex audio.

Clearly, this approach is open to additional radical improvements via audio-cues and video-sequences, etc. Even more elaborate scenarios are possible, such as sending a 3D-model of the destination with audio links, where customers are able to take a walk around the location. If the real location has an IP-infrastructure then the customer could even speak to the contact-center staff while at the destination.
9.6.2 Agents

- Personal Agent and Active Context Memory
  The Personal Agent acts as a proxy that is able to receive events from different sources such as when the user is invited to participate in communications by means of SIP. The Personal Agent is further divided into a server (persistent) part of the user’s communicating capabilities and a client part. The client allows a user to query an agent, which represents a location. The agent again has incorporated behavior to initiate Location Requests and VoIP calls. The Active Context Memory is used to store knowledge about users, such as their identities, their associated badges, accounts for messaging, voice mailboxes, personal devices, etc. It is currently used as an associative database, where the contents are interpreted as logic clauses. These logic clauses are interpreted by an inference engine, which results in the triggering of communication services through the Personal Agent.

- Contact-Center
  The Contact-Center Agent is able to handle incoming calls that originate from the telephony network or with a session request via SIP for calls that originate from the Intranet or Internet. The Contact-Center Agent decides what to do with these requests and therefore represents the core functionality of this application, while it delegates specific tasks to other agents.

- Device(s) and Service(s)
  The badge is, in this case, an active personal electronic device that has an identity, which we can use to determine the identity of a user of the system. This badge’s identity is detected by a Badge Receiver, which can be placed virtually anywhere. The Badge Receiver passes this information on to the Badge Agent, which makes it available to the other Agents. The Info Display is a public information device. The display is capable of displaying information as directed by its agent. The Location Agent represents a physical location and communicates to other Agents when events happen at its location (see further Figure 9-20).

- Infrastructure
  The Agent Server provides for the communication between Agents and is based on XSP carried on top of text based communication provided in our agent platform [3]. The Web-server (not shown in the figure) provides a means for web-access to the Agents. The Admin Agent allows for administration of the Agents.

9.6.3 Scenarios

Below we present a few scenarios that we have implemented and explored. These are far from the only scenarios that are possible and we are only at the start of research regarding such new applications. This basic infrastructure enables us to investigate which applications are possible and which applications make sense with respect to use of the Active Context Memory as augmented memory in personal mobile communications. We are still exploring what are the best design and architectural choices and why (or perhaps equally important why not).

9.6.3.1 Web, Active Objects and VoIP

We choose to use a Web-based GUI, which allows the integration of active objects (for instance Java applets). Manipulation of such active objects was programmed to result in new actions.

By associating agents with a specific HTTP URL, direct communication with agents via a browser and applets was straightforward. Typically, a new action would generate a new view, which could potentially involve other objects. A
choice presented by a personal agent to a user asking how to deal with communication in a certain context (for example, in or out of the office or at home) could be intercepted by the Active Context Memory and result in a suitable choice based on the previous behavior. The Active Context Memory then added this answer to its knowledge about the user’s communication behavior patterns. Another example would be the exchange of objects between users, by direct manipulation.

In addition, actions could be triggered by other events due to the context-aware functionality. While only proximity and location were used in the prototype, many other contexts are possible. Any of these actions or manipulation of objects could result in new multimedia sessions, such as playing a voice message from the voice server, or establishing an interactive voice and video session between two parties. Even more importantly, this would not be the result of a pre-designed program, but rather emergent behavior enabled by the Active Context Memory. The underlying signaling mechanisms that are used in the prototype are:

- communication between software agents (XSP via the text based communication provided by our agent middleware[3])
- invocation of multimedia sessions (SIP [59])

As a result, the user interface is entirely interactive with active objects and multimedia, and offers a view of the communication space that was described in Section 4.3. In the scenarios below, the XSP regarding agent communication is listed in the text where applicable.

9.6.3.2 Messaging and Voice Communication

The user receives an account from the system administrator, which results in the creation of a Personal Agent and the Active Context Memory, plus an advertisement of an HTTP and SIP URL. Web access to this URL prompts the user for a password in order to be able to use and configure applications (as shown in Figure 9-21). First, the client agent applet is initialized and presented in the web browser, it then registers with the personal agent and requests user details. The Personal Agent then replies with the actual values of the password and the user’s identity. In this example, the user already received an identity and password from a system manager who entered with the Admin{istrative} Agent and created Personal Agent, during which it obtained the information from

![Communication Space Diagram](image-url)

Figure 9-22. Messaging and Voice Communication
the Contact Center Agent. The Personal Agent thus returns the required information to the Client Agent, which in turn now can display an applet for querying the user for the correct identity and password (as shown in Figure 9-21). At this stage the user is able to browse, read, compose, send and receive messages, monitor the presence of fellow workers, and utilize VoIP-based services, such as calls or voice mail.

Figure 9-22 shows how this works. When A tries to reach B, the Voice Gateway sends an INVITE to a user with that phone number at the SIP server. The SIP server resolves this to the actual SIP URL and locates the user locally or, in case the user has moved, passes the invitation to the SIP-server where the user moved. Eventually the SIP server where the user is registered is contacted, in which case the Personal Agent starts acting on the invitation based upon the action that the Active Context Memory Agent advises. The user who has left the office has programmed a rule to forward messages to SMS. The invitation is thus redirected to the Voice Server (which acts as an answering machine) in order to record a voice message that is translated by text-to-speech and sent, as an SMS-message, to a mobile phone number. In this case, no action is required from the Personal Agent and nothing is sent back.

When the user goes logically off-line, this event is forwarded to the Contact Center agent, which being a mobile interactive space will update its list of present entities and forward this event to others. When the user is on-line and the Personal Agent receives an invitation for a speech conversation, this event results in a reply back to the Personal Agent to accept the invitation (provided the inviting party is not in the list of parties that are screened). In this example, the SIP-server actually forwards the invitation to the Personal Agent, which forwards it to the Client Agent allowing A and B to start their conversation.

9.6.3.3 Location Information

Subscription to a notification service alerts the user to the presence of another user at a specified location. Note that these notifications are interpreted and transformed into messages representing the event.

The important consequence of this is that the Active Context Memory may invoke any kind of multimedia communication including a VoIP session, triggered by these events. Figure 9-23 shows how this works. User B needs to contact User A and would benefit from knowing when A appears in a certain location L. The Client Agent provides the dialog (applet) to enter a user name and location from where a notification is requested when that user appears that location. The request is then sent to the Personal Agent. Then the Personal Agent then forwards this to the Location Agent with whom the Personal Agent subscribes for an event. When A appears in a certain location L a notification is sent by the Badge (Receiver) Agent to the agent that subscribed to it, in our case the Location Agent:

The Location Agent maintains knowledge of users and resources that are present in the location, and updates its database. Then the Location Agent notifies the subscriber of A’s presence, along with notifying the Personal Agent to B:

Finally, B’s Personal Agent must judge, using its Active Context Memory Agent, what to do with this notification. In some cases, a VoIP-Terminal or Public Information Display might be present at the location. The Active Context Agent could have stored or learnt knowledge to react to a user near a VoIP-Terminal and try and contact A. In case B is off-line, B’s Active Context Agent could still try and display a personal message to notify A, e.g. that the meeting moved to another room (see further
In our example, the Active Context Agent has no matching rules and the notification should be forwarded to B’s Client Agent. Due to the design of the agent platform, the Personal Agent reposts the notification to itself in the format that is required by the Client Agent (below) to allow for forwarding at a later time, in case the Client Agent is not present (i.e., user is off-line).

9.6.3.4 Presence Information

In addition, presence-centric communication between office workers could be modeled in a graphical application (see Figure 9-24). This approach is reminiscent of ‘Ding!’ [167] and ICQ [173].

The benefit of this approach is that it is rather straightforward and allows us to contact fellow workers by contacting their personal agents, monitor their presence, and when triggered issuing a request for a suitable type of communication (chat, VoIP, etc.). These triggers are spawned by the Active Context Memory in a similar fashion as it would able to react to the location notification subscription. This view shows the users’ presence within the scope of a Location Agent.

9.6.3.5 Context-aware Information Display

The detection of a user’s presence or absence is an event that is used as an input to determine a relevant action. In the prototype, the name of the user and the number of unread mail messages in the user’s inbox is presented on the closest Information Display (see Figure 9-25), accompanied by a sound cue (beep) to alert the user. We can easily replace the Information Display by other devices, which utilize different kinds of multimedia capabilities.

Figure 9-26 shows how this works. The Personal and Active Context Memory Agent are instructed to use public display resources when possible and have subscribed to notifications of location changes. When A appears in a certain location L a notification is sent to A’s Personal Agent also identifying the public resources in location L. A’s Personal Agent then contacts the Information Display Agent and negotiates the display of the actual content (in this case a text message as shown in Figure 9-26).

9.6.4 Conclusions concerning the Virtual Office Prototype

The important contribution from this prototype was that it demonstrated the feasibility, cost-effectiveness, and ease of deployment of an infrastructure, in which the point of service integration was moved out of the network, and application decisions were taken in the end points, employing multiple session and services simultaneously, based on any event.

The conclusions from early versions of this prototype were that KQML [180] as a knowledge query and manipulation protocol achieved very little in addition to what Java RMI [194] has to offer in terms of modelling the semantics of the communication between agents. We found it easier and generated considerably less overhead to base the design on Voyager [211] or DAEMON [3]. In addition, we found KIF [46] to be very expressive, but lacked suitable properties for object-oriented modelling of
the behavior that dynamic local behavior, this was more easily and simply expressed in Prolog and invoked from Java [179].

These experiences also pointed out limitations in the design of the service knowledge for further automatic extensions of the application behavior. Any changes in the topology of the entities and their respective goals, would require reprogramming service behavior. Clearly, while is acceptable in static applications, their value and validity is very limited. Furthermore, an Administrative agent was needed to set up the application. Its existence can be explained to some extent in that such a managerial tool is always needed at some level in any network. However, in early versions of this prototype, there was no other means for the agents to agree on the task in question, unless the application operator configured the agents accordingly, using this administrative agent. Hence, it unearthed an important conclusion, that any automatic emergent behavior extending the capabilities of this environment should adapt for a user or adapt to a changing environment when new agents become available, offering new ways to deliver services, this was beyond the design of this early experimental service architecture. Therefore, a redesign of the agents extending them with an ACM and Java classes for DAEMON [3] enabled to exchange service behavior modelled as MSK, removes these limitations, and the ad-hoc addition of a display result in the dynamic printout of the message shown in Figure 9-25, without requiring any reprogramming or intervention from a network operator.

In summary, this section described our experiences from building using a prototype of a Contact Center. Early implementations of the architecture unearthed the need for MSK and XSP in order to bring about an extensible and management-less system and provides further corroboration of my hypothesis put forward in Subsection 1.3.4.

9.7 Multimedia Service Delivery

As noted earlier, it was presumed by many that lots of bandwidth was needed to provide multimedia services. We carried out a set of experiments to determine if this assumption was true. These are described in the following subsections.

We prototyped a mobile client with a Web-based client (see Subsection 9.7.3 and 9.7.4 on page 160), where the mobility was supported by Mobile-IP and we were able to roam between LAN-segments and hand-over between wireless-LAN and GSM [69,74]. Obviously, the infrequent
address advertisements in Mobile-IP cause a second worth of packets to get lost. A more successful approach would be to apply alternative strategies for signaling and real-time audio with RTP. This will require adaptation of the VoIP client. This same approach can address radio links with heavy packet loss, this arrangement may clearly benefit from the use of an Active Context Memory that will allow the user interface to adapt according to the communication conditions.

9.7.1 Software Architecture

Figure 9-27 shows the general architecture of a mobile station. IP packets with compressed headers are transmitted directly over the radio’s link protocol. On top of IP, there is mobility management using Mobile-IP. SIP handles the invitations to multimedia sessions using TCP/UDP. Once the end-stations have negotiated the parameters of the session, samples of digitized sound are stored in the sound buffer from which the speech CODEC fetches its data. Speech encoding may also involve encryption, where packetized compressed voice is encrypted and then transmitted using RTP. In the reverse direction, when a packet is received from RTP (and possibly decrypted), it is handed over to the speech CODEC for decoding, and finally stored in an adaptive play-out buffer, until it is time to play it.

A Personal Agent and an Active Context Memory are shown in Figure 3-3, for the sake of completeness, in order to visualize how the agent may be used to react to various events (e.g., mobility, speech activity, etc.) and also act (e.g. initiating other sessions, send visual or sound cues to the user, etc.). Except for the header compression, which was not available at the time of our experiment, this architecture also applies to the prototype mobile station shown in Figure 3-5 on page 58.

9.7.2 Voice over IP (VoIP) over Fixed Access to Internet

Our early experiments [69] with VoIP over fixed access to the Internet be it dial-up or Ethernet (10-base T) showed that excellent voice quality could be attained with little or no packet loss, using various speech coders (e.g., GSM full-rate, half-rate, etc.). Latency incurred by the metropolitan area network in our test-bed [82] was negligible for the service in question (typically well below 40 msec). Other latency was incurred due to our use of large packet sizes in order to reduce overhead from IP-headers (as this work predated ROHC [17,37]). At data rates of 28 Kbps and above the packet size could be decreased resulting in an overall latency well below 100 msec and thus not noticeable to the user.

9.7.3 VoIP over a Wireless LAN Link (WLAN)

A SIP User Agent was successfully implemented and tested on various devices with WLAN access, such as notebooks (e.g., Sony Vaio), Pocket PC’s
(e.g., Compaq iPaq, HP Jornada 540), using various speech coding schemes (PCM, GSM, etc. [2]).

In the case of IEEE 802.11 (2 Mbps) and 802.11b access (11 Mbps), voice occupied very little of the available bandwidth. Packet loss would typically be at a level, which was insignificant with respect to what a speech coder could handle (e.g., Voxware RT-24). Latency incurred by WLAN access is in the order of a few milliseconds. Hence, WLAN can be regarded as ordinary Ethernet access from a VoIP point of view — see also Subsection 9.4.7 on page 139 regarding a measurement of delays incurred by handoffs.

9.7.4 VoIP over a Wide-Area Wireless Link (GSM, GPRS)

In papers [72,74,85] I describe experiments that were conducted with multimedia applications with integrated VoIP built on top of mobile computing devices with Internet access over both local area and wide-area cellular networks (GSM/GPRS) [117], where real-time speech was transmitted simultaneously with the exchange of Web content. The perceived QoS (quality, disruptions of speech) ranged from good to acceptable, and our measurements concerning latency and other relevant properties indicate that it is the design of the switched network which is responsible for bad performance (mainly latency), rather than being inherent to the VoIP solution. In addition, my papers [72,74,85] analyze in what ways GPRS/EDGE will further improve the situation and strategies for using VoIP in a wireless mobile environment.

9.7.5 Asynchronous Media

For non real-time media content, which needs ample bandwidth to deliver, there does not need to be a coupling between transfer rate and playback rate. Given a high peak-rate transmission rate to low average playback rate ratio, this means that we should be dealing with delivery of objects which can be labeled - rather than blindly dealing with delivering synchronous bits. This also means that what the communication system is really trying to do is object multiplexing and not radio link frame multiplexing - since the unit of meaningful delivery is object and not frames. It further implies that the bursty nature of computer based communication can be exploited rather than viewed as an impediment to constant bit rate delivery of bits.

As we shall see in Section 9.8 on page 163, Smart Media support modeled as application-aware agents in the end-devices and network, with the additional information regarding content along with the user’s context and communication are able to greatly reduce our requirements on the infrastructure.

9.7.6 Mobile-Aware Media Player

A calculation of the necessary storage capacity of large buffers in the mobile device shows that you could go for long periods of time without having more than a very low data-rate high-latency background service. For instance 64 MB holds 7 hours of internet radio quality audio [76,102] - so at the ~32kbps, which is available during the peak of the day (if you could use all of it) - this amount of memory could be filled in ~16000 seconds (i.e., roughly 4.4 hours). Hence by combining preloading of the memory with both low speed & burst updates it is possible to provide Internet radio quality to user for an entire day.

The capacity of a macrocell (as configured in [102]) can only supply about 64MB of total transfer during the peak 6 hours of the day (unless you want to exceed a 2% call blocking probability). This means you either have to allocate more capacity to this traffic or utilize the unused individual frames within the on-going calls or download large portions of the content in hotspots (where you have more available capacity - either because of fewer demands in this cell or because the cell has a higher data rate), or download large portions of the content in off peak periods, or have much larger buffers in the device - so you can pre-load even more content.

Dimensioning of buffers was important as it allowed us to deploy a Mobile Aware Media Player
service [82,83], where the delay was acceptable as long as you experience continuous audio. It is the user experience or perception, which is important.

Based on the results [102] regarding the unique content that is sent, each Internet radio station would send 18.25 Mb/day, assuming some day-to-day coherence the actual amount is even lower. This means that even during the peak of the day there is sufficient spare capacity to support more than three such stations even if these stations had to transmit all their content only during the peak voice hours! - All this while using only ~32kbps background capacity.

9.7.7 Hotspot Density

In this section we present data for WLAN access point density for common types of content transfer, starting with a detailed spreadsheet regarding MP3 play-out in a handheld device (assuming continuous user movement in one direction) - see Figure 3-7 and a further discussion of our prototype results in Section 9.8:

Clearly, we can support a significant number of simultaneous users with high-quality audio (MP3) with large distances between antennas. The antenna density is largely proportional to the memory size in the device, leveling off beyond 16 Mb with as many as 100 users per access point and 1575 meters between access points. H.263 video with a relatively low-bitrate is slightly more demanding in bandwidth, but there is no dramatic decrease in the maximum distance between antennas. Handovers to GPRS can provide additional download of content to compensate via low-bitrate background traffic in the order of 10-20 kbps [76], resulting in even lower densities. In addition, if we could anonymously connect to vehicles belonging to a fleet of mobile WLAN-enabled caching servers in urban areas to deliver the content, then a five megabyte MP3 file divided into fifty 100K smart media chunks, can be forwarded to and between moving vehicles. At urban speeds, and an average of 3 hops, this file will be reassembled in its entirety at the recipient’s end in approximately half an hour. No routing mechanism is required, only a time-to-live attribute for the smart-media chunks, a garbage collection mechanism, and large disks in the vehicles.

9.7.8 Near-Isochronous Communication

For near-isochronous communication, such as wireless voice chat using a G723.1 speech coder, the bandwidth offered by GPRS background traffic [102] in GSM systems is sufficient to deliver low-bitrate voice (e.g., GSM-halfrate, Voxware RT-24, etc.). However, the latency budget is used up by latency incurred due to interleaving in the GPRS air interface, which is in the order of 500 msec and time to set up per-packet paths; alternatively one could set up permanent connections and waste resources. Clearly, application awareness in the access network agent enables us to further optimize this behavior and minimize latency. On the other hand, wireless LAN is able to do fast handoffs, enabling us either use road/highway neighborhood hotspots as wireless extensions to fixed Gigabit networks (see sections starting with 10.4).

9.7.9 Analysis

In summary, bandwidth is not a fundamental problem obstructing the delivery of multimedia services over a wireless link to Internet, but rather a design issue where we must balance the cost of
delivering the services against a desired performance level and desired coverage area. Unfortunately the telecom industry continues to extend today's call model for services to third-generation wireless networks. This approach assumes reserved bandwidth in both directions. Furthermore I have presented an argument that it limits the services (as described in Chapter 3). Unfortunately for the telecom model, it does not scale well, as shown by the following example. The cost to deliver services wirelessly is proportional to the infrastructure density. Regarding the latter, the number of access points \( A_s \) for switched services can be derived from the number of users \( U \), the fixed bandwidth per service \( b_s \), the reuse factor \( R \) in relation to the total bandwidth \( B \), which is inversely reciprocal to the cell size [158].

\[
A_s \propto \frac{U \cdot b_s \cdot R}{B} \quad (9.7.9.1)
\]

This equation works adversely for large cells in wideband cellular systems, as the services consume fixed amounts of bandwidth. This calls for rethinking how the services will be delivered. When the communication between end-station is based entirely on IP, including the negotiation of sessions, we not only enable the new services, but also introduce a bandwidth on demand model for sessions. As the statistical average of used bandwidth \( b_{av} \) (e.g., messaging, voice) for mobile users is much lower than the maximum instantaneous required bandwidth \( b_{in} \) (e.g., file transfers), we may expect this model to scale much better and optimize the utilization of packet-oriented radio access points \( A_p \) than switched access points \( A_s \):

\[
A_p \propto \frac{U \cdot b_{av} \cdot R}{B} \land (b_{av} < b_{in}) \land (b_{av} < b_s) \Rightarrow A_p < A_s \quad (9.7.9.2)
\]

The above equation thus states that the number of packet-oriented radio access points \( A_p \) can be derived from the number of users \( U \), the statistical average of used bandwidth \( b_{av} \), and the reuse factor \( R \) in relation to the total bandwidth \( B \). This estimate of \( A_p \) combined with the fact that the statistical average of used bandwidth \( b_{av} \) for mobile users is much lower than the maximum instantaneous required bandwidth \( b_{in} \) and the fact that \( b_{av} \) never exceeds the bandwidth \( b_s \) required for the service in a circuit switched network implies the utilization of packet-oriented radio access points \( A_p \) to be much more optimal than the case with switched access points \( A_s \).

Any remaining concerns regarding the availability of bandwidth are again a question of design where we will have to balance the cost against desired performance and coverage, as it concerns both the network operator and the end-user. The network operator needs to understand where to provide coverage and how to get paid for delivered bandwidth. The end-users expect to have to balance their desire for services against (their consumption of bandwidth) at some expected price for this access. As a consequence, the charging model is likely to change from pay-per-service for an integrated service and cellular network access in 2G-networks (voice, messaging and even WAP) to a pay-per-byte model. This model incurs lower cost for both the end-user (who pays only for what is used) and the network operator (who only provides the connectivity), alternatively, via other access networks providing connectivity wherever the user is [66]; thus the operator becomes more and more a broker & billing agent (cf., consumer electricity sales).

Furthermore, we should question whether 3G cellular architectures are the right architectures to provide this connectivity in every situation. A key argument is that 3G-networks provide wide-bandwidth, wide-area nation-wide coverage or even across countries. On the other hand user-demand will be concentrated in urbanized areas and along transportation routes that interconnect them, anywhere else a satellite communication solution (see Section 2.2.1) is more cost-effective. Indeed, it is likely to be cheaper to give away satellite units to users in remote areas (and charge them only for their communication at the same rates as other users) rather than providing 3G-network coverage there. More over, broadband networks are being deployed connecting households, housing co-operations, and communities to the Internet in towns all over Sweden. Adding Wireless LAN
access to these networks, will instantly turn this extended network into a wireless Internet with nation-wide coverage in the same locations where 3G need to earn income in order to meet its business case. These facts are in support of an important premise of this dissertation that the wireless infrastructure will become more and more diverse and heterogeneous in future demanding applications and users to become more agile in dealing with this ensuing diversity.

9.7.10 Conclusions concerning multimedia service delivery

Multimedia services can be delivered successfully to mobile hosts. Furthermore, we have argued that bandwidth is not the problem. Nevertheless, bandwidth certainly helps, but the bandwidth that is available on today’s cellular networks for mobile telephony is already sufficient to provide some interactive multimedia services, such as allowing us to blend Web content with voice. However, what does seems to pose a serious problem for developing new applications is the call model inherited from fixed telephony into today's cellular networks. Users do not want to accept the delays currently inherent in setting up sessions using dial-up access over wireless networks. Even if these delays are reduced, we are left with separate voice and data connections - resulting in the same application limitations that have troubled ISDN.

In conclusion we may expect the cost of wide-area wireless access to the Internet to drop considerably (as compared to GSM today) and to be able to support large populations of users with VoIP-enabled applications. Also, we should refrain from incorporating complicated resource reservation mechanisms, since bandwidth is not the problem. Our conclusion is that we should focus on the development of packet based access points and mobile devices that are able to communicate via IP over a packet radio access network by simply running IP over the radio link protocol. The next chapter will show how the conclusions of chapters 2 & 3 will affect the communication industry and promote new business models.

9.8 Smart Delivery Prototype

In previous chapters and [80,84] I describe a service architecture, which allows applications to follow users, learning and adapting to the user’s communication context. A key feature of this service architecture is an eXtensible Service Protocol (XSP) that relies on meta-data describing the service capabilities and behavior of entities, such as mobile-devices, resources, or (potentially intelligent) virtual objects.

Similarly, we can add meta-data to content that is exchanged between such entities. Such an approach is taken in for instance MPEG7 where XML-formatted meta-data is used to scale and/or shape the play-out of media clips on terminals ranging from desktops with fixed Internet access to handheld devices with a low bit rate wireless access (e.g. GPRS). Even more importantly, adding meta-data to content enables agents in the service architecture network to adapt their behavior (to each other) in order to meet the criteria for specific applications. Our previous work demonstrated this smart delivery of multimedia utilizing agents running in the mobile, the access point, and the content provider [16,61,76].

We can extend this approach by creating Smart Media Packets [81] by dividing the content into
manageable chunks (the size of which is decided upon by the access network agent), to which user context and application related data is attached to the chunks as headers. This is tunneled on the application level by the agents, and reassembled at the other end. At each crossroads (places where we transfer from one network to another) there is an agent, which works as an application level router driven by the content of application & context header.

Thus, agents running in routers can inspect the additional meta-data additional which is much like the additional information which has been used to be able to recognize a the start of video image frame - thus routers know that either they should pass the whole image frame or they if they don't have resources then they can eliminate all of the packets - since delivering only part of the image frame is not useful.

9.8.1 Scenarios

Figure 9-30 shows a network overview of possible scenarios, with a low bitrate GPRS macrocell for wide area coverage, which is complemented by high bitrate WLAN hotspots. An interesting
feature is a mobile WLAN island (e.g., a taxi belong to a large fleet, equipped with a WLAN and storage capacity, to act as a mobile distribution point/proxy) that can be used by mobile users.

The user’s agents can apply their knowledge of the user’s communication to select an appropriate distribution strategy - for example dividing a movie in smaller chunks and multicasting out these to available potential access agents. As person P passes through different access areas, he/she collects and sorts the chunks in the correct order in his/her end-device and is then able to play out the movie. Users can log bandwidth conditions and locations, and exchange this knowledge with access agents in order to plan transfers of content [52,103], such as MP3 clips, voice chat, or interactive voice.

The user’s agent may also adapt to the user’s situation, by sending non-linear media, e.g. keywords or key frames for later selection of whole sequences by the recipient [65]. In addition, when we record everything the user hears or sees, we can look for familiar patterns and relate these to a communication contexts and use these as cues for further optimization of communication behavior [111]. Even more importantly we can look for particular user situations, for new applications, such as augmented memory or mixed-reality presence. E.g., filtering out key frames a 3D model of the area that we have visited can be synthesized in which our actions are mapped on a virtual robot. Visitors, visiting the area equipped with a location aware mobile device, or virtual visitors connecting to the 3D model will be able to converse with our synthesized and logged alter-ego.

Furthermore, as our agent has recorded not only our perceptual data, but also our behavior, assisted by an Active Context Memory (ACM) it is able to mimic us and in the future might act on our behalf even if we are not present.

### 9.8.2 Opportunistic Communication

A person can benefit from collecting knowledge pertaining to certain situations if they are subsequently able to use it to their advantage. For instance, the coverage of different radio networks as well as their pricing my vary greatly in certain areas. A clever strategy of how to deal with different

![Figure 9-33. Streaming Content at 64 Kbps (minimum)](image-url)
types of contents in different locations and directions (of user movement) should be accumulated over
time in an “intelligent” mobile policy table [86,161]. The requirement on an Active Context Memory
is then that this knowledge can be encapsulated (and shared between users). Figure 9-32 shows the
user interface of our media player prototype [16,61,82] and Figure 9-32 illustrates the dynamic
redirection decision based on available bandwidth for GPRS in a GSM macrocell [76] within the
context of the opportunistic download of MP3 audio in a GSM/GPRS network. The network load and
user preferences are taken into account by the ACM in the mobile device to make decisions about
current downloads in progress, the choice of access network with respect to capacity,
price/performance, expected remaining download times, knowledge about the location and
availability of hotspots (from location servers) for quicker downloads. The proxies which are located
in the operator’s access network are informed of decisions made by the mobile (or infer this when the
mobile is off-line perhaps due to lack of bandwidth for eventing via GPRS) and make decisions about
requesting new content from the application provider or forwarding content to where the user is
expected to reappear.

9.8.3 Conclusions concerning a smart delivery prototype

Together with the results presented earlier in Section 9.7, I have presented an approach to use
application-aware meta-data to deliver services to mobile users, using mobile agents that use this data
to negotiate the behavior of the applications or the resources. This approach enables a rich set of novel
services. Furthermore, our approach enables us to rethink the requirements for a Wireless Internet.

Given the high transmission rate to low average playout rate ratio, this means that we should be
dealing with delivery of objects which can be labeled - rather than blindly dealing with delivering
synchronous bits. This also means that what the communication system is really trying to do is object
multiplexing and not frame multiplexing - since the meaningful delivery is object and not frames. It
further implies that the bursty nature of computer based communication can be exploited rather than
viewed as an impediment to constant bit rate delivery of bits.

Our conclusions are that for asynchronous multimedia, instantaneous high-bitrate access enables a
mobile to receive and transmit a large amount of data during the time when they pass through a
hotspot, hence this high bitrate is more important than total coverage. Smart Media support modeled
as application-aware agents in the end-devices and network, with the additional information regarding
content along with the user’s context or communication are able to greatly reduce our requirements
on the infrastructure, in terms of coverage. Our approach enables us to tailor the infrastructure to
better suit user needs. In addition, XSP enables operators and service providers to data-mine the
agents for information, such as knowing how the services actually do operate, and thus determining
how to build the infrastructure accordingly.

![Figure 9-34. Extended Media Player Prototype]
We have created a Mobile Aware Media Player (Figure 9-31) that takes into account user movements in the network and the resultant changes in communication conditions, as well as on-going negotiation of content delivery according to the availability of (new) multimedia content on Internet media stations and intermediate media stores in the access networks [76]. The Personal (mobile) Agent uses the eXtensible Service Protocol for the necessary flexible negotiation between entities (Internet media stations, intermediate media stores, and end-users) [80]. The Personal (mobile) Agent connects to an Internet Media Station with MP3 content, which in turn diverts communication to a Content Proxy Agent in the access network. The Content Proxy also extends the functionality of the Personal Agent by sending a protocol object for streaming and playing out MP3-audio using RTSP when the user is on-line. Multimedia delivery is redirected to an optimal point of access from a user (price / performance) perspective, based on user context information: e.g., Access Network Agents notify Content Proxies in the access network of available bandwidth, and the Location Agents provide location prediction information, on the basis of which the Content Proxy the Personal Agent decides to receive content in a hot spot with 802.11b WLAN.

In addition, this approach is used to mediate between bandwidth demands for both switched mobile and remaining bandwidth for this service over GPRS in relation to other services, to the extent that we can utilize of unused frames. Furthermore, we improved our agents to recognize resource URLs from IR beacons [7,84,90] to invoke the automatic playout of multimedia content that was associated with this device. Thus we attached beacons to various locations at Brf. Bågen and demonstrated playout of multimedia that is associated with different locations in that area to visiting mobile users. These results are particularly important because they provide additional support for my hypothesis.

9.9 Mixed-Reality Bågen Prototype

Within this testbed, we created a Context- & Mobile-Aware Media Player, comprising a Personal (mobile) Agent on a mobile device - see Figure 9-34. Using the eXtensible Service Protocol for the necessary flexible negotiation between entities (agents representing Internet media stations, access

Figure 9-35. Mobile Interactive Space at Bågen
network support proxies, sensors, shared spaces, and end-users) the Personal Agent was able to: take into account user context, i.e. proximity (relative to IR-beacons), position relative to WLAN access points [138], physical position (GPS), and virtual position (relative a shared virtual space) [7], thus play media content based on the user’s context, and in addition to this, connect to a shared virtual space [77], where users are able to share multimedia objects and be aware of each other’s presence.

With this extended smart mobile media player, users could experience and exchange multimedia in Mobile Interactive Space at Bågen, mapped on top of the physical Bågen environment (see Figure 9-35 on page 167).

9.9.1 Prototype Overview

Figure 9-36 shows an overview of the prototype architecture that delivered the services which Figure 9-35 illustrates by mapping these services onto the 3D model. A mobile device was equipped with infrared sensors for emulating future types of short range radio, and combined WLAN and GPRS access. The software running on this device is the (personal) mobile agent with SIP and XSP interfaces, thus able to interact with a Bågen agent which is a Mobile Interactive Space being co-located with the SIP-server at Bågen. Co-located with this SIP Server is an Multi-Party Conferencing Unit with a Media Server attached to it (for playout of audio). The 3D model of Bågen was running on a Quake-III engine with an added external interface for the interchange of location information mainly with a Location Server (as Bågen via a Gateway (GW)).

It is important to note that at the mobile device applied sensor fusion of location (signified by the Σ symbol), users of audio-only, web-only, 3D-only, or combinations of these services could be accommodated successfully and provide them with an augmentation of their location information. Augmentation as the weakest notion of mixed-reality and implies a correspondence between virtual objects and real objects. The correspondence in the prototype pertains to location and identity and was achieved through attaching beacons [90] to real objects (a tower, fountain, and local store), and fusing object recognition (1) with location information from recognizing access points (2) or network movement from Mobile-IP (3). Recognition of real objects (1) achieved a synchronization of position relative to virtual objects. Tracking history related to the location relation (Section 5.3.4 on page 84) in combination with statistical analysis enabled us to do predictions about where the user was located in order to invoke sessions, which was further improved using a SmartBadge [106] with an accelerometer facilitating dead-reckoning. Furthermore, in addition to the virtual objects which could be visualized in 3D, we enabled the invocation of communication based events such as the triggering of background sounds (facilitated by our multiparty conferencing unit) in certain areas (indicated as dashed circles in Figure 9-35), or invocation of voice communication of users who entered different

![Figure 9-36. Mixed-Reality Prototype Overview](image-url)
areas simultaneously thereby creating “audio-tunnels” between areas in different locations of Bågen. Furthermore, location changes of users were fed back into 3D-engine and visualized as a movement of their virtual representations, but even more importantly these movements (such as proximity) were perceived as events by other entities such as Bågen’s MIS causing the triggering of other rules in their MSK.

9.9.2 User Scenario

The agents recognized resource URLs from IR beacons [7,84,90] to invoke the automatic playout of multimedia content that was associated with this device. Thus we attached beacons to various locations at Brf. Bågen (indicated as “audio spaces” in Figure 9-35) and demonstrated multimedia that is associated with different locations in that area to visiting mobile users, according to the scenario number 1-7 in Figure 9-38, showing how virtual and physical visitors to Bågen cooperate to play-out and share multimedia in their mobile devices. Furthermore, the Personal Agent can register with a Mobile Interactive Space (Group) Agent which not only forwards voice messages and real-time communication between registered participants, but it also multicasts their respective virtual positions (see Figure 9-37). Figure 9-37 shows the 3D user interface to a scale model of Brf. Bågen with on-going communication with another participant. As the Personal Agent negotiates to receive physical location information (from GPS) and virtual location information (from the Mobile Group
Agent), we have thereby created a mixed-reality system [12], in which visitors visiting the area physically can meet virtual visitors, and vice versa (Figure 9-38).

### 9.9.3 Conclusions based on this mixed-reality prototype

This prototype combined and extended the results from Section 9.5 and Section 9.6, in that it demonstrates the services that are discovered and negotiated dynamically between mobile users, virtual objects, and resources, once wireless ad hoc access to Internet has been negotiated and granted to these mobile users. Moreover, the prototype demonstrates in that with relatively low cost and ease of use, we were to extend an existing Gigabit Ethernet infrastructure with wireless access points and add to this the mobile computing infrastructure with a SIP server, agents, and beacons. Even more importantly, this infrastructure demonstrated itself to be automatically extensible as we could add sensors, users, media sources, and “audio spaces” to the Mobile Interactive Space at Bågen, without any required intervention from an operator. These results support claims made in the previous chapter about the service architecture enabling new application providers to provide services to mobile users, as end-points are enabled to negotiate services (as the media server could be anywhere). It also means that the interpretation of the objects that are discovered, can be altered relative to the MSK that is present in the device. When Figure 9-37 shows one interpretation of objects and events in this area, Figure 9-32 is another, showing the interface of the smart media-player, utilizing events to make smart decision about its communication for downloading multimedia, equally valid. Therefore, we are in the position to create new uses creating multi-purpose environment.

### 9.10 Chapter Summary

This chapter has presented an overview of possible modes of operation, which have been enabled by the open service architecture and its components. A vast range of new possibilities have been explored in this chapter, each of which shows the feasibility, scalability, and economics of building solutions based on the principles presented in the dissertation. The next chapter summarizes the results and conclusions from this dissertation.
10 CONCLUSIONS

This chapter summarizes the contributions of the dissertation. This chapter, together with Chapter 11, presents shortcomings of the solution, unsolved problems, and remaining open issues.

10.1 Overview

This research has addressed the challenge of enabling mobile communication decision making to be moved out of the network and to allow services to be negotiated and delivered between end-points taking into account the context and conditions of the user and their communication environment.

Chapter 1 examined the motivations for this research, publications that were written during its course, and listed the contributions of this dissertation. Chapter 2 presented the background and overview of the requirements and feasibility of multimedia communication on Internet (via wireless access), followed by, in Section 2.5, the necessity for a new service architecture and the impact of such a new service architecture on future communication. Chapter 3 examined the limitations in existing telecom services architectures. In light of this, an overview of available technologies and related work was presented, which in different ways attempt to remove these limitations and enable new communication, attempting to create more flexible and “intelligent” communication.

From Chapter 4 forward, the dissertation presented an answer to this challenge consisting of a service architecture and components allowing end-points to engage in communication based on multiple events of any kind and multiple simultaneous services, enabling end-points to create awareness of context and communication in a local and global context, not requiring prior knowledge of the capabilities (service profiles) of the entities involved. Therefore the solution goes far beyond the basic capabilities of UPnP to discover services and present them. A key component of the service architecture is a service modeling language Mobile Service Knowledge (MSK), adhering to the spirit of RKRL [111] in that it captures the context and behavior of communication in mobile devices, and adhering to the notation of the Semantic Web [9] regarding its compatibility with XML and logic notation, thus facilitating the examination and resolution of potential feature interaction problems. The unique contributions of Mobile Service Knowledge is that it is able to express behavior regarding invocation of communication pertaining to observed resources in its context, and is specifically designed to express behavior for multiple devices. Thus it enables a vision of swarm communication behavior involving multiple mobile devices (this was described in Chapter 8). The eXtensible Service Protocol (XSP) of the architecture removes limitations in UPnP regarding scalability and enables the discovery, eventing, and exchange of communication profiles. As the communication profiles by virtue of MSK are active, i.e., able to express the semantics of communication behavior, entities are thus able to negotiate Mobile Service
Knowledge. An Active Context Memory is the component, which collects Mobile Service Knowledge, infers new actions from this knowledge and is able to refine the semantic model of the context of the entity on the basis of observed events. Furthermore, due to their ability to observe events and active profiles in other entities, entities may aggregate this information and not only extend and refine the semantic models in their own service caches; but even more importantly, we are in the position to do data-mining of these service caches in able to learn about the performance and characteristics of our services. Hence, our service architecture has created scalable semantic mobile communication on Internet, incorporating awareness of context & communication resources, while integrating multimedia service delivery.

10.1.1 Necessary Properties of New Communication and Related Work

In Chapters 1 and 2, I presented an overview of the limitations of existing telecom service architectures in both absolute terms regarding the services that can be delivered as well as in relative terms regarding the limited potential for service growth. The problem stems from requirements for control of the services which will remain in the network, either due to the method for obtaining network access to be bundled with the service (e.g., telephony) or simply because this service architecture is the most reliable way for an operator to manage performance & resources, carry out customer administration, and most importantly provide billing & charging. The technological advances which allowed delivery of multimedia services via the Internet have had a far reaching impact. One result is that there no longer are technical reasons for maintaining the point of service integration inside the network. In Chapter 3, I presented an overview of the consequences of this revised premise, namely that with respect to personal communication in large scale (fixed or) mobile communication networks, the service architectures need to be rearchitected and can be rearchitected because we can move the point of service integration out of the network to the end-points (e.g., mobile devices), and equally importantly, that the simultaneous multi-event, multi-service, and multi-session communication which is enabled by IP, enables us to move the point of service integration to the upper layers of the communication system. Consequently, at decision points within services we have access to information from peers as well as lower layers. Therefore, these computing or communication decisions can utilize this additional information (about what the computation and communication is trying to achieve) when deciding how and when to deliver a service to an end-user. Hence, I used the term “user-centric communication” to denote this model. In addition, this relocation of the point of service integration to both the upper-layers of the protocol stack and to the end-points – enables these decisions to take into account both the context of the user and their communication; this was not possible in existing service architectures because such decisions where made inside the network where this additional information was not available. For instance consider user context, information from sensors built into the mobile device can provide spatial information or perhaps indicate the user’s current focus of communication, while other data sources might serve as “information sensors” revealing what the user is communicating or will shortly communicate. Taking these sources of information into account when making decisions about delivery enables the application to enhance the relevance of what is presented to the user and may also effect the modes in which it is presented to the user. Furthermore, the components that participate of the delivery of the service can take into account similar information about availability of communication resources and user behavior and preferences in order to make more optimal decisions regarding the use of communication resources, which otherwise would not be possible. Finally we require that new services, resources, and devices can be added to this infrastructure and be involved in both communication and decision making (when this is useful) without necessitating reengineering or upgrading of all other components, because they lack either the interfaces or information necessary. In summary, we not only require that the communication environment and the services that it offers to be adaptive, but also they must be extensible.

Further conclusions regarding the necessary properties of this new communication environment were that we must preserve scalability (necessary for being able to accommodate millions of users),
achieve reasonable network economics (providing economy of scale in addition to low cost of the infrastructure, thereby facilitating deployment through aggregated small-scale solutions), enable step-wise introduction (e.g., avoiding the need for adoption of a global standard in order for the solution to work), support the delivery of services while facilitating the on-going deregulation of roles (local access providers, ISPs, application providers, etc.), and equally importantly it must decrease the demand for management of services to such a low level that from a user's perspective it is near management-less. In summary, with respect to service, we have created plug-and-play connectivity and mobility support for all the components that need to participate in the communication.

The related work, described in Chapter 3, concerning the desired and necessary properties for the new communication environment revealed that other architectures failed to provide all these necessary characteristics. Because both ubiquitous (or pervasive computing), agent/knowledge based systems, and web services (or the Semantic Web) architectures exhibit deficiencies, relative to our objectives for user-centric computing and communication, this mandated a new approach which could integrate and extend some of these components in order to create a new service architecture framework which would enable adaptive personal mobile communication.

### 10.1.2 Analysis of Business Model Implication

In Chapter 2 I presented information about existing and future trends for providing Internet access to fixed or mobile users and gave the motivations and requirements for such communications. I then examined the impact of this new infrastructure on new business models which would be possible when services move out of the network. Trends in Internet access and the implicit move of services out of the network will accelerate the deregulation of service provisioning while fostering further diversification of roles, which hitherto had not been necessary. Bundling services and network access virtually ensured a monopolized network, and deregulation could only be more or less artificially defined between a “network operator” and “service provider” such that the later rents network resources from the former; while in practice there was no difference between the service provider and user, since the user did not have the freedom of end-to-end communication, but rather only had the service which the service provider supplied. This is especially significant since present access networks are becoming increasingly diverse and deregulated; hence users should be able to easily move between networks (perhaps even anonymously) while preserving their communication and services.

The conclusions from the analysis in Section 2.5 concerning the implications for business models provided further input regarding requirements for the new communication environment. The increasing diversity of roles accompanying deregulation means that unbundling of network resources and service provisioning will require better support for negotiating services, otherwise the effect of deregulation on the user experience will be a proportional increase in the need for user initiated configuration. With respect to business models, this dissertation foresees much more narrowly defined roles. This is already an important issue for so-called web services (e.g., ‘.NET’) which attempt to provide a replacement for centralized network based control by another system of centralized control (but with a different locus). I believe that we need to provide the support for many new business models with respect to the delivery and provisioning of communication services.

### 10.1.3 Open Service Architecture Framework

Given the combined requirements above I presented a new model (Mobile Interactive Space) and service architecture for adaptive personal mobile communication, which outlines a new framework for building systems for delivering the new services, further detailing and examining the new system: protocols, components, and the interaction of protocols, for peer-to-peer communication, enabling us to fully exploit the potential of services once they have been moved out of the network to the mobile devices. The importance of this service architecture is that it leverages Internet protocols from lower layers, and adds on top of them two other layers (the extension layer and the mobile knowledge layer)
for reasoning about and negotiation of services. In Chapter 4, I described the purpose of these layers and potential interactions between them in order to provide service mobility (while preserving personal and device mobility) and demonstrated the scalability of my solution. Unlike related work, this model supports a clear mapping between services, service negotiations, and entities, which enables parties to learn about new resources on the go; while avoiding the need to adopt a global standard for the solution to work. More importantly, given the use of a common protocol, these entities are not required to have a priori common or shared service knowledge in order to establish communication.

### 10.1.4 Adaptive Service Behavior Support

The design of the Open Service Architecture Framework which supports adaptive service behavior is comprised of a number of key components, mechanisms, and a protocol:

1. In Chapter 5, I examined the properties of Mobile Service Knowledge (MSK) and its contributions to facilitating intelligent responses to communication when attached to the mobile agents that are part of our model. An external representation of MSK is compatible with XML and the Semantic Web, but extends its internal representation with dynamic properties for describing rules and causal relations between objects (a unique property of MSK is that it internalizes observed relations between external objects in operational models) and integrates in its representation logic expressions such that the result both lends itself to reasoning and deduction for adequate responses to communication events, as well as verification of potential interaction between existing MSK and extensions that are acquired.

2. In Chapter 6, I examined the properties of an Active Context Memory (ACM) and components for aggregation, management, and inference of MSK enabling entities to exhibit awareness of other objects in their communication environment. The ACM provides knowledge management; in combination with the eXtensible Service Protocol; it enables entities (resources, users) to negotiate and act in concert and achieve a common goal within the context of a user or an application, and thus meets the necessary requirements of the communication (as described earlier in the dissertation). Equally importantly, the ACM encapsulates knowledge and reasoning (together constituting mobile intelligence) along with mechanisms for the management of knowledge and context, enabling straight-forward extension of and integration with existing SIP User Agents and can even be added to existing agents. The relaxation of requirements on the user to actively program or design their services and enabling of automatic invisible service creation and support for anticipation of reasonable communication behavior of entities reducing or preventing unnecessary claiming of communication resources to further optimize our solutions is one of the important contributions of this dissertation.

3. In Chapter 7, I described the design of an eXtensible Service Protocol (XSP), which enables peer-to-peer ad hoc communication between entities, enabling them to negotiate and extend services, and to invoke communication which potentially aggregates multiple devices. Also in this chapter I examined the internal dynamic properties and mechanisms for mobile service knowledge and how it interacts with external XSP functionality. An important contribution of XSP is that it is agnostic about the services and its general model decouples the exchange and negotiation of MSK from the internal representation in the ACM along with its reasoning and other knowledge management mechanisms, allowing these components to change and even be improved over time, without invalidating the installed base.

It is clear that this framework enables adaptive service behavior and that the architecture and components are scalable and well positioned for subsequent deployment in large-scale (mobile) networks. In addition, this framework enables the ad hoc negotiation; interchange of information; and adaptation to the availability of users, network resources, and end-devices — while enabling the delivery of the new applications.
10.1.5 Network Requirements and Properties for Deployment

Chapter 8 outlines the network requirements and properties necessary for managing mobile service knowledge and making it available to users, along with an overview of how the architecture impacts on services provisioning and roles, making it available to users (provisioning by mobile operators), where the use of packages facilitates the delivery of services in a comprehensive way. Furthermore, it describes how to meet the requirements of heterogeneous access and deregulated service provisioning. Furthermore, an analysis of the network properties of the Open Service Architecture is presented which demonstrates that it is scalable, management-less, and allows for step-wise introduction.

The service architecture enables users to automatically deal with a heterogeneous environment by explicitly addressing the management of Mobile Service Knowledge. Thus we can leverage Mobile Service Knowledge that is created and accumulated elsewhere, by storing it on Foreground/Background (knowledge) servers which may be attached to the Internet. Such knowledge servers may feature automated editors that will suggest taxonomies of Mobile Service Knowledge in combination with on-going data-mining which is done for performance analysis and verification (the later to avoid potential problems with feature interaction). The present lack of understanding of how to do such data-mining, verification, and editing does not hamper the use and deployment of the Open Service Architecture for the purposes that have been mentioned previously, but simply means that management of Mobile Service Knowledge must be studied further in order to understand how to make optimal use of Mobile Service Knowledge.

10.1.6 Feasibility Investigation and Verification

Chapter 9 presented this new model of communication, which supports multi-service, multi-session, any-event communication. In it, I have demonstrated the feasibility of multi-service personal communication that is multiplexed with end-to-end IP connectivity over wireless links in a number of key application domains. Chapter 9 also shows examples of enabled communication by discussing prototypes that implement different scenarios and present a wireless testbed and the components with which these prototypes and prototypes were built. The importance of these results was to give proof of the feasibility and practically of the theoretical results of 4, 5, 6, and 7.

In chapters 8 & 9, we show results of that new modes of communication that have been enabled by my approach, as well as improved performance of existing communication.

10.2 Contributions

The contributions specific contributions of this dissertation are:

- Characterized the necessary properties of Adaptive Mobile Communication
- Presented an overview of related work
- Description of new network requirements
- Description of new network & system level models
- Analysis of Business Model Implications
- Open (and Scalable) Service Architecture Framework
- Requirements and model for describing adaptive service behavior
- Characterized necessary (an eXtensible Service Protocol, Active Context Memory, Mobile Service Knowledge) and optional components.
- New classes of applications
- Feasibility investigation of the architecture
- Experimental results demonstrating new modes of communication and new ways of delivering services for spontaneous communication.
10.3 Final Summary

With the above results I have demonstrated that ubiquitous computing technologies can be applied to create smart-spaces which are scalable to the size of existing cellular and mobile networks. This user-centric computing and communication model involves multi-service communication that is able to respond intelligently to any event and negotiate & extend its service repertoire without requiring predefined common service concepts (that would otherwise need to be stored in the networks). Thus, by enabling services to be moved out to the mobile devices this service architecture is well suited for the deregulated and increasingly diverse (wireless) access networks to Internet. It does this while introducing mechanisms that enable application providers and network operators to query the stores of mobile service knowledge for empirical data as to how the services are used in order to learn about their relative success or to provide data that would allow simulation of service usage. This later data might be used to predict the performance or success of new services. With this dissertation I provide an important answer for (1) what applications 2.5G and later networks can support and (2) characterize the protocols, service architecture, and properties of mobile service knowledge necessary in order to enable these applications.
11 Future Work

The dissertation has shown the principles and verified these principles in prototypes, but much work remains to be done. This chapter lists a number of questions that have been left unanswered or that have been unearthed by this dissertation; based on these I have summarized some topics for continued work in these areas.

11.1 Unsolved Problems and Remaining Open Issues

This dissertation has provided answers as to how applications can be delivered to users taking into account the context of the user(s) or their communication. There are a number of remaining open issues and unsolved problems, a number of which will or should be addressed in future research. Therefore, in the course of 2002 we will further investigate how these principles and our service architecture can be applied, and conduct experiments in a few selected areas in order to gain a better understanding of how these solutions should be built. In particular we need to gain better understanding of what the implications are for fixed and/or mobile operators. This chapter presents an overview of some unsolved problems and remaining open issues subdivided into: near-term work which is expected to bear fruit in the near future and longer term work in areas which are expected to remain open issues for some time. Along with each topic is a proposal for how to conduct this research. Finally, I include a section on future prospects in this field of research, i.e., questions of more speculative nature which may become of high importance in coming years when the technological short-comings of today no longer constitute impediments to our imagination for creating such solutions.

11.2 Near Term

11.2.1 Management of Mobile Service Knowledge

The increased diversity and flexibility of the infrastructure which increasingly offers ubiquitous access to the Internet demands that users be provided with means to cope with this increased diversity and to adapt so as to exploit the flexibility offered by this infrastructure. It also raises questions as to how application providers can and will make use of the increasingly deregulated communications infrastructure; and how new business models will and can be supported.

This dissertation presented an architecture that supports the negotiation and extension of services based on meta-data about resources, user profiles, and device capabilities. This meta-data is dual in nature in that it is both descriptive and also interpretable as fully operational models of what this data represents. This dual nature is necessary in order to be able to build flexible solutions. However, we need to deepen our understanding of how to categorize, collect, store, disseminate, and make available in the general sense, such Mobile Service Knowledge (MSK) for each of the new powerful modes of communication that have been enabled by the approach.
This dissertation has also presented some mechanisms in support of the above goals, such as the eXtensible Service Protocol (XSP) which allows the use of knowledge servers, properties of MSK (packages, relations, etc.), and mechanisms in the Active Context Memory (ACM) for the management of MSK. However, it is not clear from a user’s point of view where specific knowledge may be nor is it clear how to obtain knowledge that is somehow similar to the user’s requirements. Likewise, a provider of services needs a comprehensive set of generic automated tools for mining the ACMs in order to derive specific service knowledge. In order for tools to be able to locate this knowledge and in order to be able to enforce a security policy which protects other data in the ACM from being mined by the service provider (or others), a naming and localization schema for MSK is vital.

11.2.1.1 Web Services Revisited

These problems are akin to the maintenance of ontologies of the Semantic Web or for ‘.NET’. SOAP [215] is merely a method for object access while RDF [215] is the basis for a generic modelling languages for expressing relations between web objects. Unfortunately, neither SOAP, RDF, and the use of URLs for referring to specifications residing on web servers (a practice used in the prototypes in Chapter 9) are not a sufficient answer. Hence, we should investigate solutions that do not transfer the locus of service control back into a network (as is the case for ‘.NET”), but we should use UDDI [208], which leverages HTTP, XML, SOAP, and other specifications to promote a registry of globally unique identifiers for a public list of definitions of businesses, services, and the methods for engaging firms offering these services. Similarly, we must avoid (to the extent possible) requiring the adoption of standardized concepts, as these will implicitly define a upper limit for the new services we can create.

11.2.1.2 Naming & Localization

I propose investigating a framework to facilitate the interchange and dissemination of MSK between users and providers of services. This framework will not require a registry, and should use a bare minimum of a priori defined concepts, while relying on open protocols for automatic naming and localization (i.e., classification) of Mobile Service Knowledge (MSK). This framework must scale while providing efficient support for data-mining and a comprehensive set of automated tools (as mentioned above).

11.2.1.3 Optimizations

In order to reduce the complexity of analyzing data and simplify the complexity of negotiations, packages can be used. Likewise, packages are shortcuts to negotiations of available services. By using packages we free computational resources on mobiles and short-cut the synthesizing of general semantic models by allowing a limited number of service profiles to be delivered as domain specific packages. How do we maintain our ontologies in this model? How much of this maintenance can we leave to the ACMs vs. Knowledge Servers?

11.2.2 Feature Interaction

Within the scope of the service architecture and new applications, this dissertation provided a reinterpretation of Feature Interaction and its associated problems beyond the original definition [23] In the quest for the framework we must incorporate new strategies for detecting and resolving feature interaction, and provide the detail necessary to do so beyond the general outline as described in Subsection 8.3.2 on page 122.

11.2.3 Security

In this dissertation, security has been identified as an area of key importance for establishing trust
between parties, i.e., for parties to be able to trust that MSK is not being used for malicious intent, such as “false-knowledge attacks” as mentioned in Section 6.5 on page 102. In addition, two problems arise from the more fundamental problem of what knowledge should be exposed. What knowledge should be exported and to whom [142]? Therefore, we need mechanisms to protect users from having their ACMs mined for data by parties; hence we must exercise greater control of what will be visible to specific parties. The access control lists that were presented in Section 7.5.3 on page 111 are merely an initial step towards enabling entities to enforce security policies that support dynamic sharing and dissemination of MSK between parties in ad hoc applications. These issues should be examined in greater detail and a security architecture should be proposed that is compatible with the service architecture and the mobile service management framework that was proposed above.

### 11.2.4 Smart Media Delivery

Chapter 8 presented the network requirements necessary for successful delivery of the services and Chapter 9 provides further evidence concerning the feasibility of applying the service architecture for different classes of applications. Initial results regarding asynchronous communication demonstrated that entities in the network and in the end-point(s) could dynamically negotiate the delivery of an application based on decisions made in the end-points with respect to the users’ context and/or their communication. The dissertation also pointed at the possibility for using MSK as “smart media packets” that could be used to dynamically aggregate and configure resources to deliver a certain application, and thus raised new questions regarding the feasibility, performance, and necessary dimensioning of the components for self-configuring application delivery networks through the use of mobile service knowledge meta-data [81]. For instance, when the access networks in the πG infrastructure (Section 3.7 on page 39) are populated with users who are subscribers to a service context in which they are trusted parties and willing to cooperate as store and forward nodes (e.g., a taxi company and its customers), then how long does it take before the application provider has assembled a map of the access topology and properties in for instance an urban area, or how long would it take before content fragmented and distributed in smart media packets can arrive at a certain destination?

#### 11.2.4.1 Multi-Access

More importantly, if the users have multiple access: intermittently to wireless local area networks with high bandwidth (WLAN hotspots) and continuously to wireless wide area networks (e.g., GPRS, UMTS), what optimal strategies are available to users, access providers, and application provider(s) to utilize information about local contexts to further optimize the delivery of services (be they presence, streaming media, mixed reality, etc.) or optimize the use of resources, either in the network nodes or the mobile device. Then, what sensory input is most useful? How can we optimally exploit context history? [81] presented some initial ideas and Chapter 9 provided empirical results from prototypes, but much work remains to be done either through measurements in a field trial or simulations to obtain results that can be translated into strategies and design rules for network operators (policies, dimensioning, management), application providers (agreements with operators), and system manufacturers (access points, proxies, devices).

#### 11.2.4.2 Trade-Offs

Equally importantly, there are trade-offs regarding versatility and usability. For instance, how can an entity cope with knowing about more and more agents? What are good measures for levels of usability, a service’s/agent’s/proxy’s/... relative cost in terms of memory consumption, computational resources good strategies for off-loading functions or data to the network? More research is needed in this direction.
11.2.5 Mobile Device Middleware

In Section 9.7 and other places in this dissertation, I addressed the software architecture of the mobile device. Related to this we need to provide more detail in order to reconsider the design of the components of the service architecture such that they can be adopted by industry.

11.2.5.1 Web Services Revisited

For instance, I question whether the XML formats of XSP and MSK are a priority at all, since XML is what is left when questioning the basic premise of Web Services as a priority for the service architecture — see paragraph 11.2.1.1, and XML is merely a general purpose data specification format, for which by virtue of it being general purpose there is always a translation available.

11.2.5.2 SIP

More important than XML is the integration with SIP for general purpose eventing and signalling on (a Mobile) Internet. Therefore, how to design XSP as a SIP extension [131,132] is an interesting prospect which could further simplify the design of the mobile agents. A SIP extension could be designed as a new method or reuse of an existing extension (e.g., SIP INFO method [39]) to transport XSP events. Another possible strategy is to exploit the fact that SIP [59,131] does not prescribe the use of SDP [60] and design XSP as an alternative to SDP for use when straightforward negotiation (using SDP) of multimedia session capabilities is not sufficient; when more complicated negotiations are involved, then XSP should be used. Yet another strategy may be to broaden the scope of SDP as is the focus of ongoing work in the IETF MMUSIC Working group concerning a next generation Session Description Protocol (SDP-NG) [94] and broaden the scope of SDPng.

Furthermore, we need design rules for the interfaces to ensure easy and lightweight integration of sensors and local resources, such that events from the sensors are integrated into our XSP model. Local resources can be anything from speech CODECs, a DiffServ client, the processor (an event being the device putting itself to sleep for some period of time), or the power management module. From a system vendor’s point of view, a mobile device middleware layer with plug in components for intelligent services (e.g., the SIP-UA, the ACM, etc.) makes sense, and its design should study how to minimize the requirements in terms of languages, interfaces, memory size, and complexity.

11.2.5.3 Mobile Code and Robots

Mobile Code has correctly been a popular topic of agent conferences since migratory functionality is able to execute using local information, which it otherwise would not have remote access. This simple observation has lead to my invention of MSK, which has similar migratory capabilities. A non-goal for further research is the creation of a single mobile code standard, since there are different requirements in different contexts, which motivated the creation of MSK as other designs of “mobile code” lacked the necessary characteristics. A goal for further research concerning MSK should be to clarify the design principles of service robots — see Subsection 9.3.2 on page 135 — for carrying out specific tasks remotely, and thereby making it easier to transfer a whole context and execute it in a remote environment and retrieve observations. Mobile robots enable many functions such as teleportation (see Subsection 11.3.1 below).

11.2.6 Billing & Charging

In this dissertation I referred to other results [33,66] for models of billing & charging that decoupled them from the method of access, and showed that these models were compatible with the proposed service architecture. Conversely, the service architecture and its components which will be used for delivering the services must be able to generate data which can be input to the management systems for billing and charging. For familiar services this is easily arranged. However, when new uses and new roles come into play, we will see new requirements for supporting billing and charging. Consider
for instance, if in Section 9.5 on page 142, the user who volunteers to act as a “wireless hop” were to demand to be reimbursed? In the previous subsection, if a user in a local access network hosts an application, what is the model for billing and charging? When resources are unlimited billing and charging for slices of resources seem unnecessary and as pointed out in the Introduction Chapter, billing and charging consume resources and add complexity to the network, which works adversely with respect to service growth, in which case billing and charging should be kept to the bare minimum. However, at present wireless resources are limited, so we should expect a need for billing and charging. Given this, what are good models for billing and charging relative to these (limited) resources, new services, and new business models, and what additional constraints do they put on our solutions?

11.3 Longer Term

In this section, I discuss open questions, that were examined or unearthed in this dissertation, of less well defined nature, or when the problem is well defined but our understanding of the tools necessary to solve these problems is insufficient and consequently these require further development and study.

11.3.1 Trust & Control and Ambient Intelligence

Ambient intelligence is an area of research where the central premise is that we will embed intelligence in our environment. This novel approach to communication transfers the control of what constitutes an application to the environment. In well defined cases its use is obvious, for instance when MSK representing our home device(s) is transferred to the trusted server in the hotel room we are visiting and mapped to the devices that are currently available. In public areas, while we are able to recognize its potential usefulness, its uses are much less obvious. For instance, consider using our wireless infrastructure to create intelligent streets, where lighting is adjusted based on the location and presence of people to save energy; even this simple examples raise a number of questions concerning trust & control. In addition, should the behavior of one person be allowed to affect or be visible to another person? Even if we can avoid interference between users and gain acceptance for a specific model of trust and control, what is useful and how do we avoid being interrogated incessantly by intelligent entities? How does the individual deal with trust and control? How do we escape the arduous task of having to program in detail what other people/devices/objects might see? The simple solution would be to reject the notion of ambient intelligence services, but there is no clear division between what constitutes ambient intelligence and today’s location based services. With service growth, questions related to ambient intelligence and trust & control will continue to crop up. Therefore, the questions related to ambient intelligence merit further investigation, beginning with a categorization of the research issues, proposed methods, and desired results.

11.3.2 Machine Learning

Subsection 6.2 on page 92 discussed the use of machine learning in order to improve the behavior of the ACM over time relative to the purposes of improving communication, using a number of techniques that are already available at present. Clearly, further study regarding initial values S, F, and R and their updates (as per Subsection 6.3.8 on page 98) is needed. In the longer term, we must answer a number of follow up questions, the answers to which are important for future service growth:

What are really the upper bounds for machine learning? How smart can a mobile get? Will these devices acquire real high-level knowledge and become aware, especially since mobile artifacts can have continuous access to all our sensory input? With respect to virtual presence, which was addressed in the dissertation and in the previous subsection: what are the limits for what the Active Context Memory can achieve when it has continuous access to the user’s behavior, to learn it, to mimic it, and to store patterns of our behavior. In addition: How do we present this digital personality and content to mobile users? Regarding the meaning of our communication: What communication
makes sense, how do we synthesize/achieve this? In addition, how do we measure the quality of the semantic models: what knowledge is useful (i.e., improves the model’s ability to act correctly)? How do we ensure the behavior improves to meet our expectations? How do we measure or distill our expectations?

Answers to these questions are important for future services and in order to make progress regarding the questions in the previous section. In this field of research there is a vast and expanding portfolio of techniques, which are fairly well documented in the literature. Thus, it seems most important to understand the questions and to formulate a possible approach that can use some selected techniques.

11.4 Future Prospects

In this section I discuss possible directions for the future of this field of research and the prospects of what might be enabled once the technology (e.g., the speed of computations or communication) no longer limits our solutions. Therefore, this section is inherently speculative in nature. However, experience has taught us that we continuously underestimate the speed at which technology advances take place simply because we observe one area at a time and our predictions fail as advances in several areas interact to create leaps in progress, and hence we may find that this future is already with us.

11.4.1 Cognitive Internet

As a logical consequence of the above, I envisage the creation of a “Cognitive Internet”, which is populated by cognitive entities. In fact, communication resources become abundant, ubiquitous, and ephemeral to the extent that we become mainly concerned with what is at the other end. Therefore another interpretation of cognitive Internet is the “Disappearing Network” — you will only notice the network when it does not function.

11.4.2 Mobile Brains

In addition to mobile robots and machine learning, I envisage the creation of mobile brains – while robots which are executing explicit or implicit behavior according to our specifications (however complicated these may be) a mobile brain is a step beyond; it will consist of independent cognitive processes without any explicit or implicit specifications for behavior being necessary or even available. The obvious limitation is that its behavior cannot be predicted. Given the ability of the ACM to synthesize MSK with built-in semantics consisting of meta-contexts and clusters of mobilets in response to the sensory input that the mobile device, the ACM will generate MSK in response to the sensory input to the device on which it is located. With current predictions of improvements in increase of memory capacity, processor speed, and power consumption (or production) in combination with on-going miniaturization of the electronics allows a tighter integration with our bodies such that an increasing number of sensors are able to continuously monitor and register sensory (movement, temperature, humidity, acceleration, etc.) and perceptual data (odor, images, and audio), we can indeed foresee the integration of our behavior and indeed of ourselves in the feedback loop of a cognitive system, as pioneered today with wearable devices such as the SmartBadge [106]. As the ACM generates new MSK and meta-contexts when it has continuous access to our sensory and perceptual data, discarding portions of MSK which do not contribute and reinforcing others that do according to the machine learning mechanisms that were discussed in Section 6.2, the result will be a system which will duplicate our behavior and be able to generate the responses that we do, ultimately enabling us to upload our behavior to the Internet to a mobile brain. This brain is mobile, as it is at liberty to move via the network and use the digital communication infrastructure and may act as our proxy, but is even more likely to integrate us in the digital world [128,146]. These conclusions are well aligned with those from two essays [87,88] in which I examined the properties and limitations of knowledge, consciousness, and machine intelligence, concluding that these properties can be assigned to matter (in line with the conjectures found in [109] on the ability of matter to give rise to consciousness), given that we must relinquish our ambition to gain full insight in the processes that
When we combine this with ways to integrate electronics with living brain tissue [149], this creates possibilities to help for instance people with disabilities to overcome them, as the electronics could be trained to second guess the user and stimulate nerves that may have become dysfunctional. In other cases, integration of electronics and living brain tissue has been suggested for the purpose of not only empowering users but ultimately suggesting that computing and communication devices may be our progeny [107].

Although at present wireless devices as brain implants are felt to belong to a distant future, it is the task and duty of this dissertation to point out that the MSK as a knowledge representation in combination with the reasoning and machine learning support in the ACM, are early but important contributions for enabling mobile brains and point at these possibilities as topics for future work.

11.4.3 Teleportation

A third prospect is that based on other observations regarding what may become important not only to people but also to society as a whole; is the need for teleportation, bearing in mind the cost of the steadily deteriorating traffic conditions in expanding urbanized areas (e.g., Silicon Valley, The Netherlands, or the Tri-State region of New York, New Jersey, and Connecticut [162]). [192,193] show that regarding the mobility of the Dutch population in 1998 the average travel distance per person and per day by car was 90.4 kilometers for drivers with a total population of over 115,000 vehicles, of which a decreasing number is for business and transport of goods (3,070 kilometers out of 16,300 kilometers). Most likely less than 3,700 kilometers require the transport of physical goods. The total cost of rush hour traffic jams, road repair, and loss of productive hours will force a reevaluation of traffic and transport. In Silicon Valley companies encourage telecommuting as rush-hour traffic conditions are becoming unbearable. Thus, in similar densely populated and technically highly developed areas the obvious conclusion will be to find a substitute for travel corresponding to more than \(\frac{16,300-3,700}{16,300} = 77\%\) of the travel budget, i.e., that which does not involve physical transport, and the obvious conclusion is to substitute physical travel for digital travel because the environmental cost of this digital transfer is negative (i.e., it is pure gain). At present, teleworking is an option and a lot of resistance remains to the idea simply because there are alternatives. Unless we abandon growth as a premise for our economies (which we are very unlikely to do) or emigrate from our planet (for which we lack the resources) we will need to change our attitudes and embrace the obvious: to embrace teleportation. Bearing in mind, what the previous two subsections had to say on the subject, I dare to say that the future is bright, albeit digitally so.

---

1. “Machine” in [109] did not refer to a machine in the modern sense, but rather to matter, hence the author was labeled to be the first “materialist”. 

References


[29] Paul Castro and Richard Muntz, Managing Context Data for Smart Spaces, IEEE Personal Communications, October 2000


[44] I. Fröroth, “More than Power down the Line”, Licentiate Thesis, Department of Teleinformatics, Royal Institute of Technology (KTH), March 1999, ISSN 1403-5286, ISRN KTH/IT/A VH-99/02--SE.


[107] Gerald Q. Maguire Jr., “Computers are our progeny”, Future Of Computer Science Symposium, (FOCSS), 8 June 2001, Graz, Austria


[142] Yasuyuki Tahara, Akihiko Ohsuga, and Shinichi Honiden, “Mobile agent security with the IPEdito...language”, Proceedings of the fifth international conference on Autonomous agents (AGENTS’01), MAY 2001


[164] Bluetooth is both a short-range radio link technology as well as the name of the industry consortium that supports it. Technical specifications are available from http://www.bluetooth.com/.


[167] Ding! is a presence application from Activerse Inc. (Available from http://activerse.com)


[169] GPRS Specification RLC/MAC, GSM 04.60 (GSM), EN 301 349 (ETSI).


[172] HSB, the largest Swedish organization of tenant-owned housing, and Bredbandsbolaget (B2), a new telecommunications company specializing in the provision of broadband technology to offer broadband Internet access to 350,000 households. Joint press release on August 23, 1999.

[173] ICQ is a presence application from Mirabilis (ICQ Inc. (Available from http://www.icq.com)


[177] The Internet42 Field Trial was conducted during 1997 and 1998 in Brg. Bågen providing 100 Mbps Ethernet-accesses to Internet to several hundreds of households for 200 SEK/month (source: Symposium “Internet Access till hemmet med Ethernet”, Älvsjö, Sweden, November 1999.)


[179] Jasper is a bi-directional interface between programs written in Java and programs written in (SICStus) Prolog [189]- Swedish Institute of Computer Science (SICS).


[185] The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications (Available from http://www.omg.org/).


[198] Telia (the largest Telecom Operator and ISP in Sweden) Press release regarding offering broadband Internet access to 35,000 households of Stockholmshem (a tenant-owned housing organization in Stockholm), from on September 9, 1999.

[199] Telia City Services (TCS) is a broadband network solution with an ATM backbone, commonly used to deliver IP-connectivity. Households are offered an ADSL-access, such as in Växjö (Available from http://www.araby-dalbo.com/internet/)

[200] Telia Globalcast Internetworking AB (http://new.globalcast.se)

[201] Telia HomeRun is a public Wireless LAN (IEEE 802.11b) access service to the Internet rapidly expanding (world-wide) from 140 access points in 2000: <Available from http://www.homerun.telia.com/>


[206] TINA-C Service Architecture Version: 5.0, Date: 18 June 1997. (anonymous)


[211] Voyager™ is a Java-based platform from ObjectSpace Inc. (Available from http://www.objectspace.com) for distributed object computing.


## Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>Third Generation Partnership Project</td>
</tr>
<tr>
<td>AAA</td>
<td>Authentication, Authorization, Accounting</td>
</tr>
<tr>
<td>AAAF</td>
<td>AAA Foreign</td>
</tr>
<tr>
<td>AAAH</td>
<td>AAA Home</td>
</tr>
<tr>
<td>ACM</td>
<td>Active Context Memory</td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>AIN</td>
<td>Advanced Intelligent Networks</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>AV</td>
<td>Audio/Video</td>
</tr>
<tr>
<td>B-Channel</td>
<td>Voice/Data Channel (ISDN)</td>
</tr>
<tr>
<td>B-ISDN</td>
<td>Broadband ISDN</td>
</tr>
<tr>
<td>bps</td>
<td>bits per second</td>
</tr>
<tr>
<td>Brf</td>
<td>Bostadsrättsförening (...)</td>
</tr>
<tr>
<td>BRI</td>
<td>ISDN Basic Rate Interface</td>
</tr>
<tr>
<td>CAMEL</td>
<td>Customized Application For Mobile Network Enhanced Logic</td>
</tr>
<tr>
<td>CCS</td>
<td>Common Channel Signalling System (#7)</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CE</td>
<td>Customer Equipment</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>CH</td>
<td>Corresponding Host (Mobile-IP, same as correspondent node)</td>
</tr>
<tr>
<td>cHTML</td>
<td>compressed HTML</td>
</tr>
<tr>
<td>CMA</td>
<td>Call Management Agent</td>
</tr>
<tr>
<td>CMAC</td>
<td>CMA Client</td>
</tr>
<tr>
<td>CMAP</td>
<td>CMA Protocol</td>
</tr>
<tr>
<td>CMAS</td>
<td>CMA Server</td>
</tr>
<tr>
<td>CN</td>
<td>Core Network</td>
</tr>
<tr>
<td>COA</td>
<td>Care-Of Address</td>
</tr>
<tr>
<td>CODEC</td>
<td>COder, DECoder</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CPL</td>
<td>Call Processing Language</td>
</tr>
<tr>
<td>CRTTP</td>
<td>Compressing IP/UDP/RTP Headers for low-speed serial links (RFC 2508)</td>
</tr>
<tr>
<td>CS</td>
<td>Circuit Switched</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>CSCF</td>
<td>Call Server Control Functions</td>
</tr>
<tr>
<td>CT</td>
<td>Communication Terminal</td>
</tr>
<tr>
<td>CTI</td>
<td>Computer Telephony Integration</td>
</tr>
<tr>
<td>CTRP</td>
<td>Compressed RTP</td>
</tr>
<tr>
<td>D-channel</td>
<td>Data (signalling) channel in ISDN</td>
</tr>
<tr>
<td>DAEMON</td>
<td>Distributed Ad-Hoc Environment for MOBILE Networks (Agent Platform)</td>
</tr>
<tr>
<td>DAG</td>
<td>Directed Acyclic Graph</td>
</tr>
<tr>
<td>DAML</td>
<td>DARPA Agent Markup Language</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defense Advanced Research Projects Agency</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DIA</td>
<td>Direct Internet Access</td>
</tr>
<tr>
<td>DIAMETER</td>
<td>Enhanced RADIUS (RADIUS * 2 = DIAMETER)</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Service</td>
</tr>
<tr>
<td>DTM</td>
<td>Dynamic synchronous Transfer Mode</td>
</tr>
<tr>
<td>DTMF</td>
<td>Dual Tone Multiple Frequency</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Versatile Disc (next generation of optical disc storage technology)</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data-rates for GSM Evolution</td>
</tr>
<tr>
<td>ERAN</td>
<td>EDGE Radio Access Network</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FA</td>
<td>Foreign Agent (in Mobile-IP)</td>
</tr>
<tr>
<td>FAQ</td>
<td>Frequently Asked Questions</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GENA</td>
<td>General Event Notification Architecture</td>
</tr>
<tr>
<td>GGSN</td>
<td>Gateway GPRS Support Node</td>
</tr>
<tr>
<td>GPRS</td>
<td>General purpose Packet Radio Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communications</td>
</tr>
<tr>
<td>GSN</td>
<td>GPRS Serving Node</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>HA</td>
<td>Home Agent (in Mobile-IP)</td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>HLR</td>
<td>Home Location Register</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Mark-up Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>ID, id</td>
<td>Identity</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>IDL</td>
<td>Interface Description Language (CORBA)</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGSN</td>
<td>Internet GPRS Support Node</td>
</tr>
<tr>
<td>IIOP</td>
<td>Internet Inter-Orb Protocol</td>
</tr>
<tr>
<td>IKE</td>
<td>Internet Key Exchange (IKE) (RFC 2409)</td>
</tr>
<tr>
<td>ILS</td>
<td>Internet Locator Service</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messaging</td>
</tr>
<tr>
<td>IMPP</td>
<td>Instant Messaging and Presence Protocol</td>
</tr>
<tr>
<td>IMTC</td>
<td>International Multimedia Telecommunications Consortium, Inc.</td>
</tr>
<tr>
<td>IN</td>
<td>Intelligent Network</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol (TCP/IP)</td>
</tr>
<tr>
<td>IP-MM</td>
<td>IP Multimedia Subsystem in the 3GPP Architecture</td>
</tr>
<tr>
<td>IPsec</td>
<td>IP security protocols</td>
</tr>
<tr>
<td>IPv4</td>
<td>Version 4 of IP (most widespread today)</td>
</tr>
<tr>
<td>IPv6</td>
<td>Version 6 of IP</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>IrDA</td>
<td>Infrared Data Association® (IrDA®)</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>IWU</td>
<td>InterWorking Unit</td>
</tr>
<tr>
<td>JAIN</td>
<td>Java APIs for Intelligent Networks (IN)</td>
</tr>
<tr>
<td>JINI</td>
<td>Java-based Connection Technology for Networked Devices</td>
</tr>
<tr>
<td>Kb</td>
<td>Kilobyte</td>
</tr>
<tr>
<td>Kbps</td>
<td>Kilobit per second</td>
</tr>
<tr>
<td>KBS</td>
<td>Knowledge Based System</td>
</tr>
<tr>
<td>KIF</td>
<td>Knowledge Interchange Format</td>
</tr>
<tr>
<td>KQML</td>
<td>Knowledge Query and Manipulation Language</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LAP</td>
<td>Local Access Provider</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>LEO</td>
<td>Low-orbit Earth Satellite</td>
</tr>
<tr>
<td>LIME</td>
<td>Linda in a Mobile Environment</td>
</tr>
<tr>
<td>LINDA</td>
<td>Tuple-space based shared memory computing model (Yale)</td>
</tr>
<tr>
<td>LMDS</td>
<td>Local Multipoint Distribution Service</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>Mb</td>
<td>Megabyte</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MBone</td>
<td>Multicast Backbone</td>
</tr>
<tr>
<td>Mbps</td>
<td>Megabits per second</td>
</tr>
<tr>
<td>MCU</td>
<td>Multi-party Conferencing Unit</td>
</tr>
<tr>
<td>MExE</td>
<td>Mobile Station (Application) Execution Environment</td>
</tr>
<tr>
<td>MGW</td>
<td>Media Gateway</td>
</tr>
<tr>
<td>MH</td>
<td>Mobile Host (in Mobile-IP)</td>
</tr>
<tr>
<td>MHz</td>
<td>Mega Hertz</td>
</tr>
<tr>
<td>MIS</td>
<td>Mobile Interactive Space</td>
</tr>
<tr>
<td>MMDS</td>
<td>Multichannel Multipoint Distribution Service</td>
</tr>
<tr>
<td>MMoIP</td>
<td>Multi-Media over IP</td>
</tr>
<tr>
<td>MIS</td>
<td>Mobile Interactive Space</td>
</tr>
<tr>
<td>Mobile-IP</td>
<td>IP Network Mobility Support (RFC 2002)</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MSC</td>
<td>Mobile Switching Centre</td>
</tr>
<tr>
<td>msec</td>
<td>millisecond</td>
</tr>
<tr>
<td>MSK</td>
<td>Mobile Service Knowledge</td>
</tr>
<tr>
<td>MTA</td>
<td>Mail Transfer Agent (DNS, MX records)</td>
</tr>
<tr>
<td>MX-records</td>
<td>(DNS)</td>
</tr>
<tr>
<td>NAI</td>
<td>Network Access Identifier (RFC 2794)</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>NE</td>
<td>Network Element</td>
</tr>
<tr>
<td>NW</td>
<td>Network</td>
</tr>
<tr>
<td>OIL</td>
<td>Ontology Inference Layer</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>ORB</td>
<td>Object Request Broker</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSA</td>
<td>Open Service Architecture</td>
</tr>
<tr>
<td>Oz</td>
<td>Programming Language for Distributed Systems, see MOzart</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>PARLAY</td>
<td>Open, technology independent Application Programming Interfaces managed by</td>
</tr>
<tr>
<td></td>
<td>the Parlay Group</td>
</tr>
<tr>
<td>PBX</td>
<td>Private Branch eXchange</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulation</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Data Assistant</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>PLMN</td>
<td>Public Land Mobile Network</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>POTS</td>
<td>Plain Old Telephone Service</td>
</tr>
<tr>
<td>PRI</td>
<td>Primary Rate Interface (ISDN)</td>
</tr>
<tr>
<td>PS</td>
<td>Packet Switched</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephony Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Remote Access Dial-In User Service</td>
</tr>
<tr>
<td>RAN</td>
<td>Radio Access Network</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Format (W3C)</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comment</td>
</tr>
<tr>
<td>RKRL</td>
<td>Radio Knowledge Representation Language</td>
</tr>
<tr>
<td>RLC/MAC</td>
<td>Radio Link Control &amp; Medium Access Control (GPRS)</td>
</tr>
<tr>
<td>RLP</td>
<td>Radio Link Protocol</td>
</tr>
<tr>
<td>RMI</td>
<td>Remote Method Invocation (Java)</td>
</tr>
<tr>
<td>RNC</td>
<td>Radio Network Controller</td>
</tr>
<tr>
<td>ROHC</td>
<td>RObust Header Compression</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource ReserVation Protocol</td>
</tr>
<tr>
<td>RTCP</td>
<td>Real Time Control Protocol</td>
</tr>
<tr>
<td>RTP</td>
<td>Real Time Protocol</td>
</tr>
<tr>
<td>RTSP</td>
<td>Real Time Streaming Protocol</td>
</tr>
<tr>
<td>SAP</td>
<td>Session Announcement Protocol</td>
</tr>
<tr>
<td>SAT</td>
<td>SIM Application Tool-Kit</td>
</tr>
<tr>
<td>SCN</td>
<td>Switched Circuit Network (PSTN, ISDN, GSM)</td>
</tr>
<tr>
<td>SCP</td>
<td>Service Control Point</td>
</tr>
<tr>
<td>SDL</td>
<td>Specification and Description Language</td>
</tr>
<tr>
<td>SDP</td>
<td>Session Description Protocol</td>
</tr>
<tr>
<td>SDPng</td>
<td>SDP next generation</td>
</tr>
<tr>
<td>sec</td>
<td>second</td>
</tr>
<tr>
<td>SGSN</td>
<td>Service GPRS Support Node</td>
</tr>
<tr>
<td>SHOE</td>
<td>Simple HTML Ontology Extensions</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SLP</td>
<td>Service Location Protocol</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SS[7]</td>
<td>Signalling System number 7 (CCS)</td>
</tr>
<tr>
<td>SSDP</td>
<td>Simple Service Discovery Protocol.</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>(See IP)</td>
</tr>
<tr>
<td>TINA-C</td>
<td>Telecommunications Information Networking Architecture Consortium (merged into the OMG in 1998)</td>
</tr>
<tr>
<td>UA</td>
<td>User Agent</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telephony System</td>
</tr>
<tr>
<td>UPnP</td>
<td>Universal Plug and Play</td>
</tr>
<tr>
<td>URI</td>
<td>Universal Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Universal Resource Locator</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>UTRAN</td>
<td>UMTS Terrestrial Radio Access Network</td>
</tr>
<tr>
<td>VGW</td>
<td>Voice Gateway</td>
</tr>
<tr>
<td>VASP</td>
<td>Value Added Service Provider</td>
</tr>
<tr>
<td>VHE</td>
<td>Virtual Home Environment</td>
</tr>
<tr>
<td>VLR</td>
<td>Visitor Location Register</td>
</tr>
<tr>
<td>VoiceXML</td>
<td>XML based specifications for Voice control of services</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>WCDMA</td>
<td>Wideband CDMA</td>
</tr>
<tr>
<td>WEP</td>
<td>Wired Equivalent Privacy (WEP) algorithm, which is part of the 802.11 standard.</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network (e.g. IEEE 802.11)</td>
</tr>
<tr>
<td>WML</td>
<td>Wireless Markup Language</td>
</tr>
<tr>
<td>WPAN</td>
<td>IEEE 802.15 Working Group for Personal Area Network consensus standards for short distance wireless networks</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>xDSL</td>
<td>(any type of) Digital Subscriber Line</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSP</td>
<td>Extensible Service Protocol</td>
</tr>
</tbody>
</table>