Edited by Jouni Ikonen, Matti Juutilainen and Jari Porras

5th WORKSHOP ON APPLICATIONS OF WIRELESS COMMUNICATIONS

The Proceedings of the 5th Workshop on Applications of Wireless Communications

ISSN 1456-4491

Lappeenranta University of Technology
Digipaino 2007
Message from the chairs

Welcome to Lappeenranta and to the 16th International Summer School on Telecommunications and to the 5th Workshop on Applications of Wireless Communications.

The Workshop on Applications of Wireless Communications was established to bring researchers and students working in the field of wireless communications together. The workshop is just one piece in the chain of events carried out at our summer school on telecommunications. The workshop concentrates on the potential applications of existing and emerging wireless technologies. As the applications of wireless technologies have evolved, the workshop has been modified to better meet the current demands. The summer school day concentrates on lectures on the area of service creation. The code camp deepens the subjects discussed during the lectures and the workshop by emphasizing a hands-on approach. During this year’s code camp the participants create their own services by using the FAME² platform.

The number of publications submitted for the conference was smaller than expected. This provided us an opportunity to arrange a session for doctoral students to present their current research and receive direct feedback and suggestions from the workshop audience. All the papers submitted for the 5th Workshop on Applications of Wireless Communications have been reviewed by the workshop chair and at least two other members of the program committee.

This year the program presents several aspects of applications and services in wireless environment. The keynote paper addresses the current work in service creation done by the WWRF (Wireless World Research Forum) community. The first paper of the workshop is about enabling information for the applications by using an embedded RFID approach. The next service discovery paper represents an important issue for the efficient usage of the ubiquitous environment. While we have a lot of information, context prediction allows the intelligent use of services. The community based service access paper presents an important aspect in service usage, namely the effect of social networks. The work in progress papers present some future possibilities and thus conclude the program.

The topic for next year’s workshop is Security in Mobile and Wireless Applications. We would like to invite you to participate in the arrangement of next year’s event and, of course, to participate in the event itself – see you in Lappeenranta in 2008 as well.

For fruitful conference,

Lappeenranta, 21.6.2007

Jouni Ikonen, Matti Juutilainen and Jari Porrás
Organization

General Chair

- Professor Jari Porras, Lappeenranta University of Technology

16th SSOTC Program Chair

- Dr.Tech Jouni Ikonen, Lappeenranta University of Technology

WAWC'07 Program Chair

- M.Sc. Matti Juutilainen, Lappeenranta University of Technology

WAWC'07 International Program Committee

- Professor Jari Porras, Lappeenranta University of Technology, Finland
- Professor Jarmo Harju, Tampere University of Technology, Finland
- Professor Olli Martikainen, Oulu University, Finland
- Assistant Professor Denis Trcek, Institut "Jozef Stefan", Ljubljana, Slovenia
- Assistant Professor Dario Maggiorini, Università degli Studi di Milano, Italy
- Dr. Olaf Drögehorn, University of Kassel, Germany
- Dr.Tech. Kari Heikkinen, Lappeenranta University of Technology, Finland
- Dr.Tech. Jouni Ikonen, Lappeenranta University of Technology, Finland
- Dr.Tech. Pekka Jäppinen, Lappeenranta University of Technology, Finland

Code Camp Chair

- M.Sc. Arto Hämäläinen, Lappeenranta University of Technology

Organizing Chair

- Dr.Tech Pekka Jäppinen, Lappeenranta University of Technology

WAWC'07 Local Organizing Committee

- Matti Juutilainen, Lappeenranta University of Technology
- Jari Porras, Lappeenranta University of Technology
- Jouni Ikonen, Lappeenranta University of Technology
- Arto Hämäläinen, Lappeenranta University of Technology
- Jani Peusaari, Lappeenranta University of Technology
Contents

Keynote

"WWRF White Papers: Service Architecture, Creation and Semantic Services"
Dr. Olaf Drögehorn,
University of Kassel, Germany

Session I

"Embedded RFID in Product Identification"
Tommi Kallonen and Jari Porras,
Lappeenranta University of Technology

"Service Discovery in Mobile Peer-To-Peer Environment"
Arto Hämäläinen, Jari Porras and Pekka Jäppinen,
Lappeenranta University of Technology

Session II

"Prediction of Context Time Series"
Stephan Sigg, Sandra Haseloff and Klaus David,
University of Kassel, Germany

"Community Based Service Access"
Josef Noll and Mohammad M. R. Chowdhury,
University Graduate Center, Norway

Work in Progress Session

"Information Modelling and Sharing for RFID-Equipped Objects"
Harri Hämäläinen,
Lappeenranta University of Technology

"Service Based Task Migration in Ubiquitous Environment"
Jari Porras,
Lappeenranta University of Technology

Call for Papers, WAWC’08
Keynote

"WWRF White Papers: Service Architecture, Creation and Semantic Services"
Dr. Olaf Drögehorn, University of Kassel, Germany

In this keynote Dr. Olaf Drögehorn presents ongoing service definition activities in WG2 of the WWRF. The presentation covers white papers on Service Architecture for the Wireless World, Service Creation and Semantic Services.

Biography

Dr.-Ing. Olaf Drögehorn, born in 1972, obtained his Diploma in Computer Science / M.Sc. from the University of Dortmund (Germany) in 1996. Afterwards he studied towards Ph.D. at Universities of Duisburg and Kassel with a scholarship grant of the DFG (Deutsche Forschungsgesellschaft, equal to NSF in the US).

He has worked as a project leader at the IHP GmbH (Institute for High-Performance Microelectronics) in Germany in the development of a highly scalable mobile middleware for 3G networks.

After receiving his Ph.D. in 2004 he got the position of an assistant professor (C1) at the University of Kassel. There Dr. Droegehorn is leading now the competence centre for “Mobile Middleware” within the chair for communication technology.

Dr. Droegehorn has EC and international project experience since 1999 and is now the main contact in the 6th Framework Program of the EC and within WWRF (Wireless World Research Forum) / WWI (Wireless World Initiative) for the University of Kassel. He is the co-chair of the international conference Internet Computing (IComp, since 2002), held in the USA.
EMBEDDED RFID IN PRODUCT IDENTIFICATION

Tommi Kallonen
Lappeenranta University of Technology, Department of Information Technology
P.O Box 20, 53851 Lappeenranta, Finland
tommi.kallonen@lut.fi

Jari Porras
Lappeenranta University of Technology, Department of Information Technology
P.O Box 20, 53851 Lappeenranta, Finland
jari.porras@lut.fi

ABSTRACT
Radio frequency identification (RFID) is an identification technique which has several advantages compared to traditional barcodes. One of these advantages is that RFID doesn't require a line of sight between the reader and tag which is being read. This makes it possible to embed tag inside identifiable product, where it can remain during products life-cycle without bothering the end user. There are RFID tags using several different frequency ranges and they behave quite differently while embedded into different materials.

We tested the use of different RFID tags for identifying precast concrete elements. According to our preliminary tests tags working on high frequency (HF) worked better when embedded into concrete than tags working on ultra high frequency (UHF). After preliminary tests we tested the use of HF tags based on ISO 15693 standard in laboratory conditions embedded into different materials and finally embedded into concrete elements used in a construction project. HF RFID technology proved to be reliable way for identifying concrete elements, but some practical issues need to be solved before it can be taken into production use.

KEYWORDS
RFID, Identification, embedded, lifecycle

1. INTRODUCTION
RFID is a relatively new technology for automatic identification of objects using radio waves. RFID system has two main components, the tag and the reader. The tag is attached to the object we want to identify and the reader is used to read the tags information content. The tag has at least a unique serial number for identification but many tags can have other information stored to their memory as well. RFID can be seen as a competitor to barcodes. Comparing RFID to barcodes we can see a few advantages. Because RFID is based on electromagnetic waves, it doesn’t require a line of sight between the reader and the tag. This means that the tag can be hidden inside the identifiable object. RFID tag can also contain more data than a barcode. This makes it possible for each product to have a unique identifier, not just the identifier of product group or type.

If we want to identify a product during its lifecycle, gives the embedding of tags us some advantages over barcodes, which have to be visible. In many cases it is not acceptable to have visible stickers on a product after it has been delivered to its final user. Also stickers with barcodes are not durable enough for many environments. Since RFID tag can be placed inside a product or even embedded inside products material, it can remain with the product during its lifetime without bothering the end user visually.
We tested the use of embedded RFID tags to identify precast concrete elements. The tags were embedded inside an element during its molding at the plant and they were used to identify the element during and after the construction project. The remaining of the paper is organized in following way. Chapter 2 is a brief introduction to RFID technology. Chapter 3 discusses embedding different RFID tags and what happens when we embed tags based on different operating frequency and operating principle. Chapter 4 is an introduction to our practical case, where we tested the embedding of RFID tags to identify precast concrete elements. Chapter 5 discusses the system and results of our practical tests. Finally we end this paper with conclusions.

2. RFID TECHNOLOGY

The principle of Radio Frequency Identification (RFID) has been known since the Second World War, but it has just recently become popular since the tags and readers have become affordable to many solutions. RFID systems consist of two major parts, the tag and the reader. The tag is a small and simple electronic device with antenna, memory and some electronics for controlling its functionalities. The reader is a device which creates a wireless connection with the tag and can read its serial number and possibly read/write data to tags memory. Usually the tags are used to identify different objects. They can for example be attached to a product and then the product can be identified during production anytime the product passes a RFID reader.

In a way RFID is a replacement for barcodes as it can be used to identify products quite the same way. RFID has a few features which makes it different from barcodes [1]:

− RFID doesn't need line of sight
− RFID can hold more data
− RFID can have re-writable memory

Because RFID uses electromagnetic radiation with its communication, it doesn't need line of sight between the tag and the reader. Therefore the tag can be hidden inside identifiable product. RFID tag can hold more data than a barcode. This makes it possible to assign each product an individual codes when barcodes are usually used to identify type of product, not the individual product. Some RFID tags have re-writable memory, so we can change their information content. With barcodes we really can't change the information after it has been printed the first time.

RFID systems can be classified in many different ways [2]. One way is to classify it to passive or active. Passive tags do not have a power source of their own. They rely on reader’s electromagnetic field in achieving the energy needed for their operation. Active tags do have an internal power source and they only use the reader’s field for receiving data. Active tags usually have longer read-ranges than passive tags. Another way to classify RFID systems is according to their operating frequency. The most popular frequency ranges are Low Frequency (LF, 0-35 kHz), High Frequency (HF, 13.56 MHz) and Ultra High Frequency (UHF, 869-915 MHz). The frequency range also affects the operating principle of the tag. LF and HF tags use alternating magnetic field in their communication and UHF tags and readers use electromagnetic field in theirs.
The read range of passive tags is limited by the amount of power the tag can receive from reader’s field [2]. The tag needs a certain amount of energy to its operation and it needs to receive all that energy from the reader’s field. The reader’s field strength decreases as the distance to its antenna increases. The actual amount of energy the tag receives is dependent of its antenna. So the read range is limited by reader’s field strength, the amount of energy the tag needs for its operation and efficiency of tags antenna. Also the material the readers field travels through, reduces read ranges since the field strength reduce at different rate in different materials.

In this paper we focus on embedding RFID tags. Embedding has its influence while reading the tag. This is discussed in next chapter. We also present the results of our laboratory experiments and practical tests.

3. EMBEDDING RFID TAGS

Different RFID technologies react differently to their environment. Tags that work on High Frequency (HF) frequency range work on the principle of magnetic coupling. All the data and power to passive tag is transmitted using an alternating magnetic field. The effects of the environment are minor using magnetic field compared to RFID systems using electromagnetic backscatter coupling [3]. The signal only weakens while traveling through different materials, but the tags can still be read. Metal is a major problem with HF signal. HF signal doesn't penetrate metal and also metal placed near RFID tag effects its use since metal absorbs the HF signal.

RFID systems using Ultra High Frequency (UHF) use electromagnetic field for transmitting energy and data between tag and the reader. With electromagnetic waves the effects of the environment are more severe because the environment affects the frequency on which the tags antenna responds. This phenomenon called detuning makes it more difficult to read tags contents and can even render the tag useless with designed reader and frequency. UHF signal also can't penetrate metal or water which can make the use of UHF tags difficult with certain products. The effects of different materials have been presented in a couple of publications. Foster and Burberry [4] have done some basic measurements with UHF and microwave tags attached to different products. They concluded that directional antennas are a much better choice than omnidirectional because of less disturbance to radiation pattern and return loss. Griffin et al. [5] placed UHF tags on different materials and measured gain patterns. Effects of metal and water placed near UHF RFID tag were studied by Dobkin and Weigand [6]. They concluded that the most important reason why read ranges fall near metal or water is the decrease in electric field near objects surface. Detuning is significant only when tag is very close to the object. Embedding UHF tags presents major difficulties and not too many implementations have been made. Ukkonen, Sydänheimo and Kivikoski have used UHF RFID tags with identification of paper reels [7]. While embedding an UHF tag, the tag needs to be specifically designed for the environment it will be placed in. Since the environment affects the use of HF tags much less, it is usually preferable frequency range in difficult surroundings, especially if we need to be able to read tags contents through different materials.
4. IDENTIFYING CONCRETE ELEMENTS

Precast concrete elements are molded in a factory and then transferred to construction site where they are used to construct a building. Traditionally the elements have been identified by a code which has been printed on a paper or plastic label, which can be seen on top of element in fig 1. All identical elements share the same code. There are several problems with these labels. Since an element is identified by a simple code, it doesn't actually identify an individual element. It only identifies elements type (There are also other information on these labels which could help identify individual element, but this is not always the case). These labels could come loose from element, and then you would need to compare the element with design to identify its type which might be impossible since some elements can be identical from the outside but differ in steel reinforcement inside the element. Concerning elements life-cycle the most important problem with paper or plastics labels is that they come of the element as it is mounted to its final location in a building. After this there is no way to identify an individual element. Usually there is no need for element identification after its installation, but if there is some problem with element, the factory would like to identify it to find out what caused the problem and to make sure it doesn't happen again.

![Figure 1. Test element and handheld RFID reader](image)

Identifying concrete elements through their life-cycle is a challenge, since no visible label can be left to surface of mounted element. Therefore RFID tags were chosen as identification method since the tags can be embedded inside the element. An RFID tag is embedded inside an element near its surface as it is molded in a factory. Tests were mostly done with sandwich-type wall elements which consist of two layers of concrete (inner and outer wall) and a layer on insulation in between. This structure can be seen in fig 1. The tags were placed on inside wall so they can be read in completed building without too much trouble. Since the tag can't be seen after it has been embedded, the tags were placed
on standard position to help discovering them in completed element. Specifying a standard location for tag placement was a challenging, if not impossible, job since the user should be able to read the tag during the manufacturing process and in the completed building. During manufacturing process the elements are placed in different positions during different production phases. During molding the element is placed horizontally and during storage the elements are placed vertically right next to each other. Identifying an element in storage is difficult, since only the edge of element is visible and therefore the tag needs to be placed at the edge of element or at least near the edge. The edge of element is usually not visible at finished building so we ended up placing the tag close to the edge so they can be read both in storage and in finished building. Tag placement is also an issue in finished building. Before placing a tag in a set location one should know if there will be some structure placed in front of it in finished building preventing the reading of the tag. One solution to this problem could be the use of several tags placed in different locations for identifying single element. In this case a tag could even be placed on the surface of an element for identification during the manufacturing process. For identification during the life-cycle of a concrete element we will need to embed a tag inside the element where it can remain even in finished building.

We tested the use of RFID tags for identifying precast concrete elements in three different phases. First we arranged preliminary tests to see what kinds of RFID equipment there is in the market and how they can be used for our purposes. In second phase we tested the use of RFID tags embedded in different materials. Third phase was a practical test. We embedded some tags into real concrete elements which were delivered to construction site and after installation we tested reading them to identify the elements.

4.1 Preliminary tests

In preliminary tests we had both HF and UHF tags and readers. Before the test, we molded small concrete elements where we embedded the tags. Example of test element can be seen in figure 1. The tags were placed in different locations. Some were placed near the surface of concrete and some deeper, up to distance of 8 cm from the surface. We tested two different handheld HF readers, one handheld UHF reader and two fixed UHF readers. These tests proved that HF tags worked better when embedded in concrete. The reading of HF tags worked with the better handheld reader even if they were placed 8 cm inside concrete. With UHF tags we were able to read tags placed near the surface easily with fixed readers, but using a handheld reader, reading an embedded UHF tag was nearly impossible. After these tests we chose HF tags based on ISO 15693 -standard for further tests along with a handheld reader which can be connected to a computer with USB or Bluetooth connection.

4.2 Laboratory tests

After preliminary tests we tested reading ISO 15693 tags embedded into different materials to see how they would work in different conditions where they could be used in practice. First we tested the read ranges through air as a reference and then we embedded the tags into concrete, water and ice. We tested with two different tags presented in figure 2. The
tags in figure 2 are similar in operation. Their only difference is their size and outlook. The square tag is of size 60mm*60mm and the diameter of round tag is 45mm.

Reading these tags proved to be reliable through different materials. The most important thing influencing the read distance seemed to be the tag size. Reading the larger tag always functioned from a larger distance. Measured read distances have been collected into table 1. In these tests the tags were embedded into standard depth of 40mm and then the maximum read range from the material surface was measured. In table 1 the read distance are the total distance through air and material in question.

Table 1. Successful read distances through different materials

<table>
<thead>
<tr>
<th>Tag</th>
<th>Air</th>
<th>40mm concrete</th>
<th>40mm water</th>
<th>40mm ice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round Ø 45mm</td>
<td>80</td>
<td>75</td>
<td>65</td>
<td>75</td>
</tr>
<tr>
<td>Square 60*60mm</td>
<td>160</td>
<td>130</td>
<td>110</td>
<td>150</td>
</tr>
</tbody>
</table>

The read ranges were the longest through open air. Reading through other materials didn't change the read ranges dramatically. The read ranges fell, but reading was still possible at reasonable distance. Water seemed to have the biggest effect of materials tested. These tests proved that reading HF RFID tags was indeed possible if the tag is embedded inside a concrete element. Reading is possible if the tag is embedded in the depth of a few centimeters even if the element wet or covered in ice, which is possible during winter.

5. PRACTICAL TESTS

After the preliminary and laboratory tests were completed, we tested the use of ISO 15693 RFID tags in identification of concrete elements during a real construction project. For this we needed a database for RFID and element information and software for managing that data.
5.1 Test system

The system for identification of the concrete elements is built of the following parts:
- RFID tags
- RFID reader connected to laptop computer
- A laptop computer running the user interface for managing the data
- A server with database and a Web server for user interface

![Test system structure](image)

Figure 3. Test system structure

Figure 3 represents the test system. A RFID tag is embedded inside the element to be identified. Before embedding, the tag is read and its serial number is added to elements information in the database. The database, on the server, holds all the data connected to elements and implements a web user interface which can be used on any networked computer with a web-browser. The system for these tests was really simple and was only used for adding/modifying data concerning element in question and retrieving this information. Steps for adding new data and retrieving data are presented in figure 4. The user is using a laptop with RFID reader connected through USB-port and a network connection. Adding and retrieving information happens with the help of a browser and a web user interface.

![Flowchart](image)

Figure 1. Steps for getting and adding element data
5.2 Results

In practical test we embedded 29 tags into different types of pre-cast concrete elements. The tags used were the square tags presented in figure 2. These tags were embedded into three different types of concrete elements. 9 into balcony floor element, 2 into a wall element dividing two balconies and most importantly 18 into sandwich type wall elements. The tags were embedded inside the elements at a factory and reading the tags was tested at construction site after the elements were installed into their final position and the building was almost complete. Summary of the read test is presented in table 2.

<table>
<thead>
<tr>
<th></th>
<th>Installed</th>
<th>Not tested</th>
<th>Found</th>
<th>Not found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balcony floor</td>
<td>9</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Balcony wall</td>
<td>2</td>
<td>-</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Wall</td>
<td>18</td>
<td>3</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>All</td>
<td>29</td>
<td>10</td>
<td>17</td>
<td>2</td>
</tr>
</tbody>
</table>

Finding the tags in completed building proved to be difficult. Since all the elements at the construction site were not tagged we had the find to tagged ones. Also we had a lot of problems with tag placement which explains the not tested -column in table 2. Some problems we encountered:

- With the balcony floor elements we run into trouble with the tag placement. In several cases the tag was placed at a location were a pillar was placed on top of it and we couldn't get close enough to read the tag.
- With wall elements we had similar problems. In many cases the tag was placed at a location, where it was blocked by other parts of the building.
- At construction site there was three buildings and all the tagged elements were supposed to be installed into one of them, but since there were identical elements in other buildings. Some of the tagged elements might have been switched to elements without tags and the tags were not found. This explains the not found column.

Out of 29 tags a total of 17 tags were found after the elements were installed. 10 tags could not be tested because of practical issues with tag placement. 2 tags were not found, but this is most likely explained by the fact that similar elements were placed at the same construction site into other buildings and the elements were probably switched. Most importantly no certain failed reads were discovered, we were able to read all the tags which placement we knew for sure.

6. CONCLUSIONS

In this paper we discuss the possibility of using embedded RFID tags in product identification during their life-cycle. RFID tags can be embedded inside a product where it doesn’t bother the end user visually, but the product can be identified without opening it. We tested the use of RFID tags in identification of precast concrete elements. In preliminary tests the tags which use HF frequency range proved to be more reliable than
UHF tags when embedded inside concrete. Therefore we further tested HF tags based on ISO 15693 standard in a practical case where tags were embedded inside concrete elements and reading the tags content was tested after the elements were installed into their final position at a building. Some practical issues with tag placement were discovered, but reading HF tags even if embedded inside concrete element proved to be reliable.

Embedding RFID tags gives us some interesting possibilities. Using a tag gives each product a unique id which can be used to collect and monitor data through product lifecycle. This could benefit many industries, since collecting data connected with unique product becomes much easier which should help companies follow and improve their product quality.

REFERENCES

SERVICE DISCOVERY IN MOBILE PEER-TO-PEER ENVIRONMENT

Arto Hämiläinen
Lappeenranta University of Technology
P.O. Box 20, 53851 Lappeenranta, Finland
arto.hamalainen@lut.fi

Jari Porras
Lappeenranta University of Technology
P.O. Box 20, 53851 Lappeenranta, Finland
jari.porras@lut.fi

Pekka Jäppinen
Lappeenranta University of Technology
P.O. Box 20, 53851 Lappeenranta, Finland
pekka.jappinen@lut.fi

ABSTRACT
Future pervasive computing service platforms are widely used with wireless mobile devices. In such wireless ad hoc and peer-to-peer environments, service discovery plays important role. In addition to functional service discovery protocol, efficient utilization of available network technologies improves the usability of an ad hoc communication system. PeerHood is our peer-to-peer environment, which tries to address these issues. This paper presents the service discovery process in PeerHood.

KEYWORDS
Service discovery, peer-to-peer, ad hoc networking

1. INTRODUCTION

Mobile networks are in a significant role in future service networks. Mobile service networks are not created by duplicating all Internet services, but by selecting suitable ones and by creating new ones to support them. A service platform is required to control the service creation, deployment, control and discovery to enable efficient service operation in mobile networks. The I-Centric Communications platform defined inside the Wireless World Research Forum [1] is such a platform. Tafazolli [2] named peer-to-peer and ad hoc communication as the first requirements for I-centric service platform. This paper will describe PeerHood, our peer-to-peer environment, and especially its service discovery procedures.

Service performs some task needed by the user. Mobile Services are services which are used by wireless mobile devices. Mobile communication can be based on infrastructure or it can be ad hoc communication in peer-to-peer manner. Particularly in ad hoc
communication, Mobile Services set requirements for service discovery process, placing service discovery protocols in essential position.

We start this paper by summarizing related research in service discovery area. General components of a service discovery protocol will be explained and characteristics of existing service discovery protocols compared. Subsequently, we present the general architecture of PeerHood, our Peer-to-Peer environment. The main topic of this paper is to describe the service discovery procedure in PeerHood. This discussion is presented after general description of PeerHood environment. Finally, we give our conclusive remarks.

2. RELATED WORK

Considerable amount of research and development has been done in the context of service discovery protocols, mainly for use in wired and infrastructure-based wireless networks. The best known examples of service discovery protocols developed both in the industry and academia are Lookup service of Jini [3], UPnP [4], Bluetooth Service Discovery Protocol [5], Service Location Protocol [6] and UDDI [7].

Zhu et al. [8] have created a common taxonomy for service discovery, and compared service discovery protocols according to components based on this taxonomy. In this taxonomy, 10 different components have been discovered. Naming of services and attributes can be either template-based or template-based and predefined. Division can also be done according to initial communication method - whether it is unicast, multicast or broadcast. Services can be discovered and registered either by querying or based on announcements. In the service discovery infrastructure point of view, certain protocols require service directory, whereas others can operate without a directory. State of service information can be either hard-coded or soft state. The scope of service discovery can be based either on network topologies, user roles or contextual information. Usability of services is often improved or weakened by the chosen service selection method. Automatic service selection based on application-specific service weight simplifies the selection process, but may cause selection of unsuitable services. On the other hand, manual service selection causes the user more inconvenience. Service invocation is done simply by providing service location, defining communication mechanism or even application operations. Service usage is explicitly released or based on leases. Client can keep up with the status of service by receiving notifications from the service, or by explicitly polling it periodically.

In [9], authors have discovered three approaches to service interaction. First way of interacting with a discovered service is to download the service component to the client device and execute it on the device. There’s no need to maintain connectivity after the service component has been downloaded. Alternative approach for interacting with the discovered service is through a defined interface. The interface component can be downloaded to device like in Jini, or the interface can be defined in the control messages of the service discovery process. Yet another interaction method exists. Emulation can be
Bluetooth wireless technology has been developed as an ad hoc wireless technology, and therefore service discovery has been a requirement from the start of the development of the technology. Bluetooth Service Discovery Protocol (SDP) is a part of the main Bluetooth specification [5]. Bluetooth SDP operates on Bluetooth Logical Link Control and Adaptation Protocol layer (L2CAP). SDP can be used to search for specific Bluetooth service or browse services on a specific device or on all remote devices. SDP retrieves service records for each service, including service name, service class, protocol descriptors and profile descriptors. Bluetooth SDP uses unicast as an initial communication method, although usually SDP is run immediately after Bluetooth inquiry process with discovered devices as target addresses, reaching all discoverable and connectable devices in the neighborhood.

3. PEERHOOD ARCHITECTURE

The main goal of PeerHood is to provide a communication environment, in which mobile devices act in a peer-to-peer manner [10]. PeerHood helps applications to find remote devices and services and connect to them. Device and service discovery is an essential part of such environment. In order to quickly establish ad hoc connections, PeerHood is constantly monitoring the wireless neighborhood for other PeerHood-enabled devices. Services on each PeerHood device are stored locally on device and service database. Stored information is provided on request for PeerHood applications.

In addition to core immediate personal area network (PAN), consisting devices found nearby, PeerHood is designed to enable connections to selected distant remote devices. Using infrastructure networks, e.g. Internet, home, corporate and vehicular area networks, a user can extend his network to cover all his personal services regardless of his location. This approach called Personal Network (PN) has been introduced in [11]. In PeerHood, this approach is supported with the GPRS plugin, which uses GPRS or similar infrastructure network to expand the PeerHood neighborhood.

PeerHood architecture can be divided in three separate entities. The PeerHood daemon is the component, which ensures the continuous knowledge about other wireless devices. PeerHood library is the component, which provides applications interface for operations provided by PeerHood. PeerHood library is used both in local communication between application and daemon and between PeerHood applications in different devices. PeerHood plug-ins provide functions specific to the wireless network technologies. Plugins are used by the PeerHood library and the PeerHood daemon. Relationship between different components can be seen in Figure 1, which pictures the PeerHood software architecture.
3.1 PeerHood daemon

Main operation of PeerHood is performed by the PeerHood daemon. PeerHood daemon is a background application, which takes care of the device and service discovery. Its responsibilities include maintaining a list of both remote and local services.

Communication between PeerHood-enabled application and PeerHood daemon is handled using local socket interface provided by the PeerHood library. Applications can use this interface to search for remote devices and services and to register services to local PeerHood daemon. Daemon takes care of answering to service requests.

3.2 PeerHood plug-ins

Different network technologies have been implemented as plug-ins, which can be loaded dynamically by other PeerHood components. Currently, network plug-ins have been implemented for Bluetooth, Wireless LAN and GPRS.

Bluetooth offers several choices for data communications. First of all, data communications can be established directly on Logical Link Control and Adaptation (L2CAP) layer. L2CAP is the layer, which all data transmission rely on when using Bluetooth wireless technology. On top of L2CAP, RFCOMM protocol provides virtual serial port support, on top of which either OBEX or PPP connections can be established. OBEX is designed for exchange of small data objects, for example vCards. PPP provides
point to point IP connection between devices. However, another choice for IP connections over Bluetooth exists, namely Bluetooth Network Encapsulation protocol (BNEP), which offers IP connectivity directly on top of L2CAP without the need of RFCOMM layer. This provides smaller overhead and better network formation as RFCOMM/PPP combination. We have chosen direct L2CAP operation for Bluetooth connectivity in PeerHood. We can avoid additional overhead caused by either RFCOMM and PPP or BNEP, and still offer ordered and reliable data delivery.

Wireless LAN (WLAN) and GPRS plug-ins for PeerHood both operate over IP connections. The difference between WLAN and GPRS plug-ins comes from the service discovery and connection establishment process. WLAN plug-in uses broadcast-based service discovery and offers direct connections between communicating devices without the need for additional devices. GPRS plug-in uses proxy device as an intermediary. Reason for requiring a proxy with GPRS usage is the usual practice from to provide only private non-reachable address for GPRS clients. The plug-ins have been named after the most appropriate network technology for the plug-in, although they are not strictly limited to those technologies. For example, WLAN plug-in can also be used on IP networks created by Bluetooth BNEP protocol, just by choosing the BNEP network interface.

3.3 PeerHood library

Applications use the PeerHood library both to request information from the daemon and to connect to remote services. The PeerHood library is a software component, which can be dynamically loaded into an application. It provides an interface, which has all the required functions for PeerHood operation. Applications can register their own services to a local PeerHood service database, which is advertised to remote devices on request. Also, service discovery requests to local PeerHood daemon are sent using the PeerHood library.

The library is also used to establish connections to remote services, and to transmit data between the devices. From the application point-of-view, connection establishment and data transmission process is transparent from the chosen network technology. To provide a common interface for several wireless technologies is a secondary objective of PeerHood, primary objective being the ability to monitor wireless neighborhood and to provide information about nearby devices and services. Furthermore, initial support for roaming between different wireless connections has been developed for PeerHood.

4. SERVICE DISCOVERY IN PEERHOOD

Service discovery in PeerHood is query-based, and it’s done by the PeerHood daemon. During initialization, the daemon loads configured network plug-ins, which start monitoring the wireless neighborhood for other PeerHood devices. When a new device is found, plug-in updates the device entry in the device database managed by the daemon. After that, device and service info can be requested from the device. Each network plug-in has slightly different ways to discover new PeerHood devices. Characteristics and
constraints of each wireless system have been taken into account, and the most suitable method has been selected.

Bluetooth inquiry process and Service Discovery Protocol specified in Bluetooth wireless technology provides us means for discovering Bluetooth devices running PeerHood. We can use a specific Universally Unique Identifier (UUID) for PeerHood. When a device providing this Bluetooth service is found, it’s known to be a PeerHood device. We can continue with PeerHood communication tasks. However, since the Bluetooth service discovery is strongly connected to Bluetooth specification and technology itself, it’s not feasible to use it when using other wireless technologies.

Wireless LAN plug-in of the PeerHood uses single hop broadcast messages to find out other PeerHood devices. After starting to listen for replies, PeerHood daemon sends a broadcast message to a predefined port number. PeerHood devices in the same subnet receive the broadcast message and respond to it by sending unicast reply back to the source address. The devices, which responded to request, can be added to the local device database.

The GPRS plug-in is configured to use a proxy, which forwards PeerHood device discovery requests to all connected devices. Likewise, replies to discovery requests are forwarded back to the requesting device. After receiving these replies, requesting device can add new remote devices to common device database. The GPRS plug-in is mainly intended for expanding the PeerHood neighborhood by creating Personal network [11] with pre-defined connections to favorite services. Other use is to provide supporting infrastructure-based technology for roaming between network technologies.

4.1 Service Registration

In order to be found by other PeerHood devices, PeerHood application needs to register its service to the local PeerHood daemon. PeerHood daemon maintains a database of available devices and services. As service location information, a service entry includes the port to connect. Other information registered includes name of the service and the attributes. Currently, little research has been done on PeerHood Service descriptions, but service description templates and definitions are one of the future research areas of PeerHood. Initial survey of related research suggests, that in addition to common service description templates, a system should offer predefined set of typical services, to remove ambiguity regarding the service descriptions. Services should have also user-friendly, human-readable forms, in case of manual service selection by the user. Service location protocol specification [6] defines service location information as universal resource locators (URL), which are of the form "service:"<srvtype>"://"<addrspec>. URL’s are a well known form and therefore could be a good way to provide human-readable information about the service type and location.
4.2 Service Discovery Process

After daemon has found neighboring PeerHood devices with the technology specific device discovery methods, common sequence of PeerHood service discovery can be carried out. The process of PeerHood service discovery and interaction is shown in Figure 2. To clarify the representation in the figure, the required step of device discovery is omitted. Operation of PeerHood daemon includes constant search for other PeerHood daemons in the neighborhood and sending of information requests to discovered devices. Requested information may include device info, service info, available network technologies and list of neighboring PeerHood devices already found by the target device.

When starting up, the PeerHood application (client or server) initializes the PeerHood instance by creating a local socket connection to local PeerHood daemon. This connection is later used to register services to the daemon and to request the list of found devices and services from the daemon. As soon as PeerHood server application has registered its service to the daemon in its own device, that daemon responds to the arriving service discovery requests with the updated service list responses. This is depicted in the Figure 2 with two different service discovery responses sent by the daemon of Device 2. At first, before the registration of the PrintServer application, an empty service discovery response is sent, while after the registration the response includes updated service information with Print server service.

Daemon constantly sends information requests to other PeerHood daemons found in the neighborhood. The contents and the time interval of the requests can be configured by the administrator of the device. For example, we might not need to request list of services every time we discover a certain device. Or, we might exclude the request for the list of neighbors from the information request, if we are interested only for devices and services found nearby.
After the daemon has received information from remote PeerHood devices, it updates device and service information in the local device and service database, managed by the daemon itself. The information is ready to be supplied to the local PeerHood applications. Applications can query a list of all discovered devices and services, or devices providing certain service. Daemon returns the list of locations of requested services. These requests are done using the local socket connection, established during initialization.

All the communication between PeerHood application and Daemon is done by using functions provided by the PeerHood library. Also connection establishment and data exchange between PeerHood applications is done using the PeerHood library. Daemon and library uses network plug-ins as an actual component communicating with other devices.

PeerHood has been designed to utilize extended neighborhood information, which includes the exchange of stored device and service information between neighboring PeerHood devices [12]. Neighborhood information can be used to synchronize the device databases between different devices. It diminishes the influence of failed device discovery attempts, and it can be used also to devote specific device (e.g. personal computer) to perform device and service discoveries on behalf of mobile device. This is carried out in the regular device discovery process, with requesting the list of neighbors in the information request sent between daemons.

4.4 Service Interaction

Of the service selection approaches presented by Tyan [9], we have chosen to define an interface as the method for service interaction in PeerHood. For each PeerHood service, the service discovery process results to a location of the service. Location information includes reference to PeerHood plug-in which can be used to connect to device, the technology specific addresses of the remote device, and the port for the target service. Connection establishment based on retrieved location information is done automatically by the PeerHood library. Upon a successful connection, the library returns a connected interface reference to the application. As for the service selection mechanism, PeerHood library provides both automatic and manual ones. Application developers may choose whichever is more suitable. Automatic service selection is done by ordering the PeerHood interface to connect to a desired service type. Manual selection can be done by requesting a list of service entries of certain type from the PeerHood daemon, and selecting the most suitable one from the list. By providing two different methods for service selection, we can overcome the inconvenience caused by the manual service selection, but offer a manual selection when necessary.

5. CONCLUSION

Service discovery is a fundamental part of future network environment. Wireless connectivity places service discovery in an important role. This is even more emphasized in ad hoc connections, with no possibility for permanent centralized directory for services.
We have developed PeerHood, a system for ad hoc networking, providing connections using different wireless network technologies: Bluetooth, Wireless LAN and GPRS. To address the important issue of service discovery in wireless networks, a device and service discovery process has been implemented as one of the central activities of PeerHood. This process considers different requirements and possibilities of wireless network technologies, and provides the most suitable ways of discovering other devices and services. PeerHood also includes a common interface, which can be used to interact with found services, regardless of the underlying network technology.

We have chosen to offer both automatic and manual service selection. This enables both convenient automatic service selection of common services, and on the other hand, more exact manual service selection, if service type requires accurate consideration of service attributes, or if the service attributes include ambiguous information.

REFERENCES


ABSTRACT

Context awareness is a key feature of modern computing applications, allowing application behaviour to be adapted to context. The power and quality of context-aware applications can significantly be increased by not only considering past and present contexts for adaptation, but by also predicting and reacting to future contexts. This paper deals with fundamental challenges of context prediction. After an introduction to context prediction and a clarification of the basic terms associated with it, the paper proposes solutions concerning prediction algorithms and describes an architecture for context-aware computing, including context prediction, we have developed. The findings described in the paper are illustrated and verified by simulations using different types of context data and different prediction algorithms.

KEYWORDS
Context awareness, Context prediction.
While the authors in [1, 9, 12] use the term context prediction to describe an operation that infers future contexts from past and present contexts, [10] uses this term in order to describe the automatic triggering of actions when some context becomes active, while the authors of [13, 14] use it to describe the process of inferring context from sensor outputs. In our understanding, context prediction can be used by applications to react to events that will likely occur in the future. The information base on the user context is therefore expanded by context prediction. The cost for this additional information is an increased error probability of the predicted context information. It lies in the nature of prediction that the reliability of a predicted element is typically not comparable to observed present or past events. While the impact of weak reliability may differ from application to application, this is definitely the most serious drawback to context prediction. The choice of the prediction algorithm is therefore serious. Popular algorithmic approaches are Markov approaches [1, 15] or statistical methods as the autoregressive, moving average or ARMA algorithms [16, 17, 18]. An alternative approach to context prediction is to utilise alignment methods in order to predict future contexts. Prediction algorithms are intertwined with other components in context prediction architectures. Without a general architecture, the uses of context prediction are restricted to single applications. An architecture provides a standard interface that can be utilised by several applications. Since context prediction is especially useful in mobile and ubiquitous environments, the architecture should support a distribution of components as well as discovery of services and components and dynamic adding and removal of components. With Foxtrot we introduce an approach that serves this need. The remainder of the paper is organised as follows. Section 2 introduces basic concepts in context prediction that are utilised throughout our work. Section 3 proposes an architecture for context awareness that includes modules to solve context prediction tasks. In section 4 we utilise various prediction algorithms in several simulation scenarios in order to obtain simulation results that back our discussion in previous chapters. Section 5 summarises our results.

2. FOUNDATIONS OF CONTEXT PREDICTION

In the following sections we introduce basic concepts that are related to context prediction.

2.1 Concrete Context Time Series

Context prediction algorithms predict sequences of future patterns. This is done with knowledge of the sequence of observed user contexts. We refer to these sequences as context time series.

**Definition 1**

Let $i, j, k \in \mathbb{N}$ and $t_i$ describe any point in time. A time series $T$ is a non-empty, ordered set of context elements $v_i$ that are attached a timestamp $t_i$. We write $T_{i}^{j,k}$ in order to express that the attached timestamps of the context elements in $T_{i}^{j,k}$ are in $[t_i, t_k]$. The context elements in a time series are further grouped to time series elements.
Definition 2

Let $T$ be a time series and $t_i \in \mathbb{R}$ be a timestamp of any one context element $v_i \in T$. A time series element $\xi_i \in T$ consists of all context elements $v_i \in T$ that share the same timestamp $t_i$ ($\xi_i = v_i$ and $\text{Timestamp}(v_i) = t_i$).

Another property typically found in time series recorded in realistic scenarios is that time series elements contain only part of the context information at any point in time. Since different sensors most probably generate an output value at different points in time, a time series in typical realistic cases is no generic construct.

Definition 3

A concrete time series $T$ is a time series where any time series element $\xi_i \in T$ may contain one or more context elements of any combination of context sources.

With concrete time series (cf. Fig. 1), operations on these time series that are otherwise straightforward become more complicated. Assume for example that we want to find similar context patterns in two concrete time series. Typically, no subsequence of sufficient length is found where the aligned sub-sequences consist of equal sensor outputs considering number and type of the sensors.

![Figure 1: Illustration of a concrete time series.](image)

2.2 High-level and Low-level Contexts

Context has several levels of abstraction that are commonly described by the notions high-level context, low-level context and raw sensor data [19, 20, 21]. Although a commonly accepted definition does not exist, higher context representations are typically symbolic while lower representations are numeric.

We classify the context abstraction level by the amount of processing applied to the data. The output of a sensor is considered as raw data since it most probably needs further interpretation. Sensor outputs are differing even for sensors of the same class. This is because of distinct representations or accuracies of the sensed information. A pre-processing of raw sensor data is necessary so that further processing is not influenced by special properties of the sensor itself. We refer to this pre-processing as the context acquisition step. Raw data is normalised to a representation utilised by all further context processing steps.
The low-level contexts of two sensors of the same class measured at the same time in the same place are identical with the exception of a possibly differing measurement accuracy, provided that both sensors are in good order. In order to obtain high-level context information, the low-level contexts are further processed in the context interpretation step. Possible operations are aggregation with other low-level contexts, interpretation, data calibration, noise removal or reforming of data distributions.

There is no limit to the level of abstraction. High-level contexts may be further processed to again receive high-level context information.

Table 1 exemplifies raw data, low-level contexts and high-level contexts for the keyboard sensor. The same key pressed on an English and a Russian keyboard (raw data identical) might result in different low-level contexts due to an alternative language setting (acquisition procedure). In the Cyrillic layout the letter ‘ё’ is obtained while it is the letter ‘z’ for the English layout.

However, for German keyboards the letters ‘y’ and ‘z’ are exchanged compared to the English layout, hence leading to the same low-level context even though the raw data is different. Furthermore, different context interpretation procedures may lead to distinct high-level contexts (office occupied or writing).

<table>
<thead>
<tr>
<th>High-level context</th>
<th>Low-level context</th>
<th>Raw data</th>
<th>Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>writing</td>
<td>z</td>
<td>0x79</td>
<td>keyboard</td>
</tr>
<tr>
<td>writing</td>
<td>Ё</td>
<td>0x79</td>
<td>keyboard</td>
</tr>
<tr>
<td>writing</td>
<td>z</td>
<td>0x7a</td>
<td>keyboard</td>
</tr>
<tr>
<td>office occupied</td>
<td>z</td>
<td>0x7a</td>
<td>keyboard</td>
</tr>
</tbody>
</table>

Table 1: Examples of high-level context, low-level context and raw sensor data for an exemplary sensor.

2.3 Context Prediction

The task of a prediction algorithm is to find a context sequence in an UbiComp environment that most likely describes the continuation of the observed time series.

Definition 4

Let \( k,m,i \in \mathbb{N} \) and \( t_i \) describe any point in time. Furthermore, let \( T \) be a concrete context time series. Context prediction is the task of learning a prediction function \( f_{t_i} : T_{i,k,i} \rightarrow T_{i,m+1,i} \) that describes the behaviour of the user at time \( t_i \).

This definition combines all parts that constitute the problem of adaptively computing future contexts in dynamic ubiquitous computing environments [22].

2.4 Data Abstraction Level for Context Prediction

Since context has several data abstraction levels, we have to decide on which abstraction level the context prediction task is to be implemented. As raw sensor data is not yet in a normalised representation, several complications in context processing would be introduced if prediction was based on raw sensor data. We therefore suggest not to utilise raw-data for context prediction.
In the literature context prediction is usually based on high-level context information (see for instance [1, 9, 15]). These authors first interpret the low-level context to obtain high-level context information. Afterwards the prediction is based on the observed high-level contexts. However, prediction based on high-level context information has vital restrictions due to a reduced knowledge about the context itself [23]. We therefore propose to base the prediction procedure on low-level context information (cf. figure 2).

![Figure 2: Context prediction based on low-level context elements.](image)

We discuss issues on the context prediction accuracy that originate from the utilisation of data at various abstraction levels in the following sections.

3. AN ARCHITECTURE FOR CONTEXT PREDICTION

We introduce a general architecture for context prediction that was presented in [24]. This architecture is integrated into the Framework fOr ConteXT awaRe cOmpuTing (Foxtrot). Foxtrot follows a service oriented design with components that can be distributed between various mobile devices.

4.1 Foxtrot

We present an architecture for context awareness that is inspired by our main research interests and that serves as the basis for current and future research projects related to context awareness. We focus on a modular design and extensibility. This includes the possibility to easily exchange modules, if possible even at runtime.

The architecture is supposed to be executed in a distributed environment. Consequently the deployment of components of the architecture, discovery and network communication are required. To meet these requirements, we adopted a service oriented architecture approach and implement all functionalities as interacting, loosely coupled services. These services were built using the FAME² [25] framework. FAME² enables adding, removing and updating of services at runtime, which eases the maintenance of the system. Additionally it allows us to integrate several service discovery technologies and communication protocols to access arbitrary services from any other device reachable.
FAME$^2$ is a middleware that abstracts from the details of various computing platforms and provides a common interface regardless of the computing device to applications and services. FAME$^2$ is currently running on a multitude of computing platforms including mobile phones as the nokia communicator, smart phones as the MDA IV, pda’s as the iPAQ and tablet PCs, notebooks and desktop computers running a windows or linux operating system.

The processing of context information in Foxtrot is a hierarchical process that is composed of three main parts. The first is the context acquisition part in which the sensor information is acquired, the second is the context processing part, in which the context information is further processed and the third step is the context usage part, in which the context information is utilised by applications and services. This process is illustrated in figure 5.

![Figure 5: Architecture for context prediction.](image)

The context processing part of this process may be further divided into various processing steps.

4.1.1 Sensors

We consider sensors as atomic information sources. Basically, a sensor is some piece of hardware or software that provides information on the environment. Humans or animals are not considered sensors but might trigger and influence sensor outputs. We distinguish between hardware sensors and software sensors. Hardware sensors are physical devices that react to stimuli from the physical environment. Hardware sensors might for example measure the temperature, the light intensity or the humidity. Further hardware sensors are a computer keyboard or a mouse that monitor the user input or a fingerprint reader.

Software sensors are applications that react on software generated stimuli and that output a software generated notification describing this stimuli. Examples of software sensors are a calendar, an address book or an application the user is interacting with.

Similar to the Context Widget approach proposed by Dey[19], we encapsulate sensors by software components that provide a uniform interface between sensors and higher layer components (cf. figure 6).

![Figure 6: Conceptual design of a context broker.](image)

We will refer to these encapsulating components as context brokers and assume that context brokers can be arbitrarily distributed on computing devices in the UbiComp
environment. Pre- eminent components a broker consists of are a driver for the sensor encapsulated and a buffer containing sensor output values. The context broker hides the sensor specific interface from higher architectural layers and provides a standardised interface to enable dynamic sensor discovery and removal.

Since sensor measurements are stored in the buffer, the context broker can publish a sensor output at any time requested by simply providing the last value measured or by extrapolation.

4.1.2 Context processing

A context processing step processes context information. It is constituted from one or more processing modules, a common input interface and an output buffer that serves for communication with other modules in the architecture. Typical processing modules are noise removal, data calibration, context reasoning and context prediction. Processing steps can be distributed among devices in the environment. Figure 7 depicts an exemplary processing step.

![Figure 7: Conceptual design of a processing step.](image)

4.1.3 Applications

Basically, the only requirement for applications that acquire context information from Foxtrot is the Foxtrot-interface. Applications subscribe to processing steps to receive the computed output of these processing steps in an event-based communication.

To ease the development of services that comply with FAME\(^2\) an eclipse-plugin has been developed at our chair.

4.2 Context Prediction Architecture

We insert the context prediction architecture between the context interpretation and the context acquisition layers (c.f. figure 2). Observe however, that due to the general definition of input and output data as context time series, the architecture could also be utilised for context prediction on high-level context elements (i.e. between the interpretation and the application layer). In section 5.2 we utilise this property in order to compare high-level and low-level context prediction schemes in simulation scenarios.

The architecture is constructed from four basic modules; a learning module, a context prediction module, a rule base and a context history. The learning method, the rule base and the context history are implemented together in one processing step, while the prediction module constitutes the main part of the second processing step. Furthermore, these modules interact with further processing steps in the context acquisition layer and the context interpretation layer (cf. figure 8).

All functional entities in the architecture are represented as processing steps which can be distributed on devices in the environment.
Context prediction based on low-level contexts is considered as one pre-processing step. Therefore, context interpretation, feature extraction and other pre-processing operations are not managed by the context prediction architecture. An illustration of an exemplary context data-flow can be found in figure 9.

Low-level contexts obtained from context brokers in the context acquisition layer may be further processed by context processing steps before they are utilised by applications.

The context processing steps might process low-level contexts in a context prediction component or might interpret low-level contexts to high-level contexts or also process high-level contexts.

The figure shows that not all data is utilised by the prediction component.

**4.2.1 Context history**

We require the context prediction architecture to implement a continuous learning procedure that adapts to possibly changing user habits. We therefore require a context history that is constantly updated. The context history represents the 'memory' of the prediction algorithm which is utilised to extract rules describing the typical behaviour of
the user. The context history time series is fed by all context sources available in the context acquisition layer (cf. figure 8).

For each sensor attached to the architecture we track a sensor description and a time series of low-level context elements in the context history. The maximum length of the tracked time series is dependent on user preferences, implementation decisions or on memory constraints of the user device.

The context history is closely coupled with the learning method since it provides the input for this method. In order to protect the communication link between the learning method and the context history we integrate the context history in one processing step together with the learning method. The context history can be considered as the interface of the learning method to the observable environmental features.

4.2.2 Rule base

The rule base contains the rules that the learning algorithm constructs to guide the prediction module. The language in which these rules are represented is dependent on the prediction algorithm that is to access the rules for prediction tasks. Some examples are:

- The time series of observed context elements
- Typical patterns in the time series
- Transition probabilities related to context changes or sequence changes
- A function describing the time series evolution

The rules in the rule base might have a weight attached in order to express the importance of the rule. We propose to keep the rule base on the user device since the rules generated are closely related to the context evolution the device experiences. Consequently we integrate the rule base together with the context history and the learning method in one processing step.

4.2.3 Learning component

The learning method has to support continuous learning in order to adapt to changing user habits or environments. It therefore constantly monitors the context history and eventually uses some or all of these time series to construct and update rules that are then stored in the rule base. Since we do not propose a specific implementation the only requirements for the learning method are the interface specified by the context history and the description language for rules in the rule base which is defined by the type of the prediction algorithm.

We describe the ability of the algorithm to adapt to a changing environment with the learning frequency. The learning frequency describes how frequently the method screens the context history in order to find new prediction rules. With a higher learning frequency hence the processing load of the learning module increases. However, an algorithm with a high learning frequency will more often update its rule base. A rule base that is more frequently updated will in general describe the current user habits more accurately.

4.2.4 Context prediction component

Context prediction is done by the prediction module. It accesses the rule base and the momentary sensor data provided by the context sources in the context acquisition layer.

The task of the context prediction method is then to find the context sequence that most likely follows the most recently observed context time series. We had to take a design decision regarding the acquisition of recent context information in the context prediction
Either the prediction component utilises the context history of the learning component or it provides a separate history of recent contexts on its own.

We propose to provide a separate history of context evolution in the processing step of the prediction method since the prediction algorithm might require another representation for the observed context time series and since it might utilise only a subset of all available context sources, as some context sources might be considered unimportant or unreliable.

By providing a second history of contexts in the prediction component, this prediction-algorithm specific complexity is hidden from the learning component. Furthermore, with this design various prediction components might utilise the same learning component.

4. SIMULATIONS

In order to compare context prediction algorithms we apply a Markov approach, an ARMA algorithm and an alignment prediction approach to various sets of sampled data.

5.1 Prediction of wind power samples

In order to utilise wind power, wind farms are built that consist of a multitude of wind turbines. The amount of power available from these parks is fluctuating and dependent on the wind power. Since the power supply system is not capable of dealing with these fluctuating power curves, methods that predict the wind power are required to schedule the power expulsion of other power sources the in order to balance the power level.

5.1.1 Simulation Scenario

The data set contains wind power samples from a wind farm in Germany and was recorded from February 2004 to April 2005 in an hourly fashion. We utilised 3/4 of these samples as training data and the remaining part for the simulation.

We apply two context prediction algorithms to the data set. Namely a Markov approach of order 6 and the alignment prediction method. Both algorithms have their context history size restricted to 6 hours. We complete several simulation runs in which we change the prediction horizon of the algorithms. Prediction results for prediction horizons ranging from 1 hour to 20 hours are obtained.

Additionally, we utilise an ARMA approach with an unconstrained context history as upper bound and a modified version of the ARMA algorithm (ARMA-100). For ARMA-100 the context history size is restricted to 100 hours. To give an estimate on the accuracy of the prediction we calculate the RMSE and BIAS for every algorithm as

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - d_i)^2}{n}} \quad BIAS = \frac{\sum_{i=1}^{n} \left| p_i - d_i \right|}{n}.
\]

In these formulas, \( p_i \) depicts the predicted value at time \( i \) while \( d_i \) is the value that actually occurs at time \( i \).
5.1.2 Simulation Results

The results of the simulation regarding the RMSE values are depicted in Fig. 10. Both ARMA and Markov methods are nearly identical for short prediction horizons while the alignment prediction approach performs worse. However, with prediction horizons that exceed the context history size (6 hours and more), the alignment prediction algorithm outperforms both the Markov and the ARMA-100 implementations. Still, it does not reach the same accuracy as the unrestricted ARMA approach.

However, while the ARMA approach has access to the complete context history, the alignment approach utilises, apart from the training data, only a context history of six hours. With a context history restricted to 100 hours the ARMA-100 approach is worse than the alignment approach for higher prediction horizons.

Results for the BIAS metric are similar. The major difference to the RMSE results is that for higher prediction horizons the accuracy of the Markov approach becomes nearly identical to the ARMA-100 algorithm.

Summarising, we have observed that the alignment approach performs well on the data set for high prediction horizons, whereas for low prediction horizons the Markov and ARMA-100 approaches may be preferred.

While these results have to be backed by further investigations, they show that the alignment prediction approach is well suited to extract plenty of information from short context histories.

This property makes the alignment prediction approach especially well suited for ubiquitous computing environments. Since the user behaviour might be described by a series of short characteristic context patterns, it is essential that the prediction algorithm is capable of extracting a maximal amount of information from these patterns. We will investigate this property in further studies on sampled user-context time series.

![Figure 10: RMSE-values on the wind-power data set.](image-url)
5.2 Location Prediction

We study influences of varying levels of context abstraction on the learning capabilities of high-level and low-level context prediction schemes on a sampled GPS trajectory of a mobile user. The sampling process lasted for about three weeks. The sampling hardware consists of a mobile phone and a GPS-receiver. A python script running day and night on the phone is used to obtain the GPS-information from the GPS-receiver. This GPS data was utilised for our study.

5.2.1 Simulation Scenario

The simulation consists of three weeks of GPS data from a mobile user. Every 20 minutes a GPS sample is taken. For times when no GPS is available (e.g. because the user is indoors), we assume that the last available sample best approximates the current position of the user.

For the low-level context prediction we directly utilise the GPS-samples. For the high-level context prediction we define a total of 36 high-level locations as for example 'Home', 'Bakery', 'University', or 'Market'. The high-level locations are specified by a GPS-centre-location and a radius. A default high-level location named 'Outdoors' is applied when no high-level location matches a GPS sample.

As prediction algorithm we implement an alignment prediction approach [26] for low-level and high-level context prediction. The algorithm has a context history of 5 hours and a prediction horizon of 6 hours. All simulation specific parameters of the algorithm are identical for high-level and low-level prediction.

For evaluation of the accuracy we utilised the RMSE and BIAS metrics.

5.2.2 Simulation Results

We apply three simulations with varying amounts of learning time. In the first simulation no prior learning is allowed. In a second implementation, we utilise the first half of the data for learning purposes and in a third implementation, 3/4 of input data.

Table 2 depicts the RMSE and BIAS values for high-level and low-level context prediction schemes.

<table>
<thead>
<tr>
<th></th>
<th>Low-level</th>
<th>High-level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>BIAS</td>
</tr>
<tr>
<td>No learning</td>
<td>0.7589</td>
<td>0.5759</td>
</tr>
<tr>
<td>½ learning</td>
<td>0.5883</td>
<td>0.3461</td>
</tr>
<tr>
<td>¾ learning</td>
<td>0.5879</td>
<td>0.3456</td>
</tr>
</tbody>
</table>

Table 2: RMSE and BIAS values for high-level and low-level context prediction schemes after various learning times.

For low-level context prediction a learning prior to the simulation improves the simulation result. However, the difference between the second and the third simulation is minor, since only few new patterns are observed in between. For high-level context prediction we observe that the overall result is worse than for the low-level case. While various reasons may account for this property [27], the learning gain is insignificant.

We observe this property also in figure 11.
In figure 11, the fraction of the low-level RMSE-values divided by high-level RMSE-values is depicted. We observe that the accuracy of the high-level context prediction scheme is above the low-level context prediction scheme for the first four days. This is due to the lower complexity of the high-level contexts. However, in the pace of the experiment, the low-level context prediction outperforms the high-level context prediction due to the higher information entropy of the low-level contexts. The depicted results contain two incursions at about the seventh and the fourteenth day. These are the weekends, where the GPS trajectory of the user significantly differs from his everyday trajectory. The low-level context prediction scheme is obviously better capable of coping with the new context patterns. Furthermore, the second incursion is stronger since the low-level context prediction scheme better utilises the information provided in the first weekend.

We further observe that the learning in advance of the simulation has a significant impact on the accuracy of the low-level context prediction scheme. For the BIAS-metric we have observed similar results.

Finally, we analyse the amount of memory required by the low-level and high-level context prediction schemes. For every typical context pattern observed, the alignment prediction algorithm stores a time series in a so-called rule base. We count the number of time series in stored and the number of context values in all time series (cf. figure 12).
Note that these figures also influence the processing load since the algorithm processes every single time series in the rule base in order to find alignments most similar to the observed time series.

We observe that the ratio of time series and context elements for low-level to high-level prediction schemes increases over time. After a simulation time of 20 days the low-level context prediction scheme stores roughly 1.5 times the amount of context values as the high-level prediction scheme. The number of context values per time series is more or less constant but approximately 1.1 times higher for low-level context prediction.

We therefore conclude that the low-level prediction scheme has higher memory and processing requirements although the simulation parameters are equal.

5. CONCLUSION

The ability to predict future contexts allows to improve the proactivity and quality of context-aware computing applications. In this paper we have introduced the basic principles of context prediction. Furthermore, the paper has described three different types of algorithms that can be used to predict future contexts. Foxtrot, an architecture for context-aware applications based on the FAME\textsuperscript{2} middleware, has been introduced, and we have described how context prediction is implemented in this architecture. Finally, simulation results have compared different context prediction schemes and their accuracy. Additionally, we presented simulation results of the novel alignment prediction algorithm on a realistic context prediction problem. The investigations described in this paper show that a lower data abstraction level is beneficial to the context prediction task, since it contributes to a better adaptability of the prediction algorithm. However, the memory and processing requirements of low-level context prediction are higher compared to high-level context prediction schemes. Furthermore we have demonstrated that the alignment prediction algorithm is able to extract plenty of information from a short context history.

ACKNOWLEDGEMENT

We would like to acknowledge partial funding by the German ”Bundesministerium für Bildung und Forschung” (BMBF) in the framework of the Wireless Internet and mik21 projects.

REFERENCES


COMMUNITY BASED SERVICE ACCESS

Josef Noll and Mohammad M. R. Chowdhury
University Graduate Center, UniK
Kjeller, Norway
josef.noll@unik.no, mohammad@unik.no

ABSTRACT
Ubiquitous computing and the pervasive Internet have enabled service access in every situation. However, adaptation to the user needs is purely handled, and service specific security implementations are only found for specific services. This paper presents an approach to combine the I-centric and service centric world based on a semantic description of user preferences and user relations connecting to proximity and remote services. Based on personal, corporate and social identities security requirements are defined for handling the service access. A prototype using over-the-air key distribution demonstrates the capabilities of the suggested approach.

KEYWORDS
i-centric, user-centric, SOA, profiles, personalisation, NFC

1 Introduction
Current developments in service delivery have the focus in two areas: (i) Mobile Service Delivery and (ii) Semantic Service Delivery. Current reporting from The World Factbook states 1.5 to two times as many mobile users as Internet users for developed countries like UK, France and Germany and roughly three times as many mobile users as Internet users in China [1]. Taking into account that mobile users are available 24 h/7 days a week as compared to an average PC usage of just above 2 h/day[2] shows the importance of mobile service access [2].

Mobile service provision is hampered from the limited interface capabilities of a mobile phone, thus needs to perform service personalisation, including adaptation to personal preferences, terminal and network capabilities. This is one of the key challenges for mobile operators [3], and subject of current activities in the Wireless World Research Forum. The forum has initiated an I-centric service architecture [4], which puts the communication behaviour of human being in the frequent interactions with objects in their environment.

Machine understandable Web and Web Services are the goals of developments in Semantic Web and Semantic Web Services [5]. Semantic Web is seen as the next generation of the Internet where information has machine-readable and machine-understandable semantics. Semantic Web Service implementations are seen as an extension of the Service Oriented Architecture (SOA), allowing a.o. for automatic service composition.

This paper presents an approach to combine the I-centric and service centric world based on a semantic description of user preferences and user relations connecting to

\[137.3 \text{ minutes/day for male users and } 134.2 \text{ minutes/day for female users}\]
proximity and remote services. It explains the principles of the I-centric and service centric world in sect. [2] with a focus on service delivery in mobile/wireless environments. It then introduces in sect. [3] an identity architecture, covering private, corporate and social identities. Based on the privacy requirements of a user in a certain context, it will then in sect. [4] provide a concept and a prototypical implementation, followed by the conclusions.

2 I-centric versus service centric approach

This section will focus on the commonalities and differences in a user-centric (or I-centric) and service centric approach. The difference between both approaches is historical, where a service centric architecture was introduced to let services communicate with each other. The I-centric approach, postulated by the Wireless World Research Forum (WWRF), is based on the transition of access delivery to service delivery [3]. Current rule-based algorithms become too complex when handling user context and preferences, thus asking for new mechanisms allowing dynamic adaptability of services. The service centric world was introduced based on service level agreements (SLA) between trusted partners. In a more dynamic service provisioning world, as envisaged in a Semantic Web Services environment, privacy and security become key issues [6]. Our approach is to take advantage of developments in both worlds, using the security and privacy mechanisms of the I-centric world and combine them with the semantic representation of data as known from the Semantic Web (Services) World.

2.1 I-centric vision

Access provision was the key issue in first and second generation mobile networks (1G, 2G-networks), while service provisioning is key in 3G and Beyond-3G networks. "Systems beyond 3G" will provide personalized wireless broadband access, and will incorporate mobile and wireless access methods including e.g. Wifi, WiMAX [3]. Offering personalized broadband wireless services across networks, both national and international, will require new ways of service interconnectivity.

The key challenge in personalized broadband wireless service access is the handling of user preferences, context, devices, and connectivity. The European project ePerSpace introduced personal service delivery in the home segment, based on user profiles and preferences [7]. Experiences from this and similar projects showed that managing and updating preferences is a tedious work and that users often disagree with the selected services resulting from a rule-based decision engine. While the home is a rather controlled environment, with trusted and known constellations of devices, service delivery in the mobile/wireless world is more complex and thus even more complex to handle. Louis V Gerstner, Jr of IBM said: Picture a day when a billion people will interact with a million eBusinesses via a trillion interconnected, intelligent devices.
Pervasive systems does not just mean computers everywhere; it means computers, networks, applications, and services everywhere. The report from the UK Technology Strategy Board [8] pointed out that the high-added value comes from:

- **Always on** - availability of the right content at the right place and time.
- **User-centric** solutions - simple and practical person-oriented solutions.
- **Invisibility** - numerous, casually accessible, often invisible computing devices.
- **Intelligence** - removing the cognitive load through devices with embedded sensing and processing capabilities.
- **Increasing productivity** - market value propositions: saving time, saving money.
- **Life-enhancing** - penetration of technology into mainstream mass market applications.
- **Innovation** - using technology in ways that empower people to work, live, and play in radically new ways.
- **Omnipresent** - embedded into everyday devices and objects all around.
- **Ubiquity** - everyone and everything connected to an increasingly ubiquitous network structure.

To build these types of personalized services is a challenge to the system design as well as the user personalisation is easy and intuitive. Several authors suggest that personalisation might be supported by "learning" profiles handling the preferences of the user, the "presence" (where is the user, what is he doing), and the social/community characteristic of a user [3],[4],[9]. Such systems would become too complex to be handled through rules, thus reasoning is seen to be a more appropriate matter to handle what to present to the user.

Personalisation is based on handling the user’s identity. Approaches for a mathematical description of identities have a long tradition. Khoshafian claimed back in 1986 the need for a 'strong support of identity', and described identities through a graphical representation [10]. The introduction of semantics and the representation in .rdf and .xml allows describing user preferences and relations to characterise the social context of the user as indicated in fig. 1 for a school scenario. Paul and Anna are members of class two of Sogn school, and their parents, here: Frank and Maria are defined through a friend-of-a-friend (foaf) based relationship. This paper connects social relations to document and service access as illustrated in fig. 2 here a Web camera connected to the classroom, and photos/videos taken by the parents. Our service scenario builds on the relation between the actors, and establishes access rights to services and documents. Further details on the selected approach are given in sect. 4.
2.2 Service centric approach

New methodologies, techniques and tools are necessary to develop and maintain services for the future that are both attractive, easy to use and cheap enough. Concepts and technologies like Service Oriented Architectures (SOA), Web Services (WS), Semantic Web (SW) and Semantic Web Services (SWS) have gradually grown up to show their viability, especially if they are used in combination. Semantic Web-based technologies are widely acknowledged to play an important role in solving the interoperability problem between applications; the usage of semantic description in the context of advanced services delivery is expected to support easy access to the services. Not only such formal and explicit descriptions enable easy service integration, but will also support exchange of preferences, profiles and context information of mobile users.

While SOA as a vision evolved well, different implementations hampered the applicability. To avoid those problems standardisation done by OASIS established a framework and drafted a reference model that a system has to adopt in order to claim compliance with the OASIS SOA specifications [11]. According to the OASIS frame-
work SOA is an architectural paradigm (model) that does not necessarily mean usage of Web Services although Web Service is a popular implementation. The SOA reference model should captures core principles and axioms of SOA and be used as a template for the SOA architecture [12]. One prototypical implementation of a Semantic SOA platform was performed in the European Research project Adaptive Services Grid (ASG$^2$), which based the platform design on the following service specific requirements, using Web Service Modelling Ontology (WSMO) [13].

- **Reuse** of existing functionality, e.g. services and infrastructure, makes use cases cost-efficient to realise.

- **Standards and Reliability** are essential for industry to adopt solutions.

- **Openness** will allow integration of additional services with as little changes as possible.

- **Adaptivity** to current environmental constraints, e.g. user preferences and user connectivity is key for user acceptance of new services.

- **Dynamic** and transparent service composition is required to adapt to the specific service requests.

- **QoS awareness** handles specific user requests, e.g. budget or time constraints.

- **Semantic Awareness** is crucial for understanding the user request, service discovery and service composition.

ASG fulfils the requirements, and extends the specifications of the OASIS SOA reference model. In spite of the big conformity to the reference model it is a great chance that interoperability problems will occur between an ASG system and an OASIS compliant SOA system. The reason is the choice of different semantic standards for describing their respective data models [14]. While a technical solution might be expected in the time frame 2009/2010, issues like privacy and protection of user requests and dynamic service level agreements between service providers might hamper the time to market. Kagal et. al. pointed out similar findings and claimed the necessity to extend Web Services with privacy and security [6]. They suggested an extending of Semantic Web Services based on OWL-S with policies, representing security requirements for service discovery and privacy protection of user requests. However, this alone does not solve all issues when it comes to an I-centric mobile user scenario, as addressed in detail in the next section.

$^2$http://asg-platform.org
2.3 Mobile service world

The mobile service world has made the move to a SOA oriented architecture. Most of the mobile services like location information are available through a Parlay X Web service interface [15]. In [16] a semantic annotations of advanced Telecom services was used to achieve exchange of roaming information on a dynamic basis. The main findings of the approach were the cost reductions in service delivery, due to reduced effort for testing and updating of Web services in a semantic service world.

Two issues remain unsolved when it comes to the usage of SOA in a mobile environment, (i) the variation of the radio quality and (ii) specific mobile services in the proximity of the user [9]. Radio is a shared resource, and the quality of the radio link is affected by user mobility, radio environment (user speed and coverage radius), application topology, and user terminal requirements. A service oriented platform builds on reliable, minimum delay and high-bandwidth connectivity, which is not achievable in mobile/wireless environments.

The service world of a mobile/wireless user consists of proximity and remote services. Examples of proximity services are admittance services or payment through contact-less cards. These services are moved to the mobile phone through Near Field Communications (NFC) and prototyped world-wide, e.g. from Mastercard in Dallas [17]. One goal of these field trials is to demonstrate interworking between wireless technologies and NFC, another goal is to address security issues like potential threats, identity, privacy and simplicity. Adding NFC capabilities to the mobile phone opens for key exchange through near field and through the mobile network, thus providing a principle way of delivering authentication information. The prototypical implementation in sect. 4 will use short messages (SMS) to distribute admittance keys, which are used for admittance to a building.

3 Identity based service access

In the real world, each of us has created his own spheres of identity. Identity is reputation: what I say about me and what others say about me [19]. My reputation is different, depending on whether I am at work, doing sports, or enjoy membership awards in a club. In the virtual world identity handling is more difficult, taking into account the dynamic service requests and privacy requirements of a user. Roccas introduced this in 2002 through the term of social identity complexity, defining a new theoretical construct that refers to an individual's subjective representation of the interrelationships among his or her multiple group identities [18].

Identity is mainly verified through an authentication mechanism. The Internet was built without such an identity layer. In the current Web2.0 discussion Identity2.0 is introduced to interconnect people, information and software. Various institutes and industries are working to provide better identity management solutions. In Liberty
Alliance members are working to build open standard-based specifications for federated identity and interoperability in multiple federations, thereby foster the usage of identity-based web services. Within this, they are focusing on end user privacy and confidentiality issues and solutions against identity theft. Another solution, Sxip has been designed to address the Internet-scalable and user-centric identity architecture. It provides user identification, authentication and internet form fill solutions using web interfaces for storing user identity, attribute profiles and facilitating automatic exchange of identity data over the Internet. To access online services, Windows CardSpace uses various virtual cards (mimic physical cards) issued by the identity providers for user identifications and authentications, each retrieving identity data from an identity provider in a secure manner.

Most of these identity mechanisms are tailored towards remote services. In this paper we focus on methods of using different identification mechanisms for the variety of remote and proximity services, thus providing an Identity management for the I-centric and service centric world.

### 3.1 Representing the Identity

The proposed integrated identity mechanism consists of certificates, keys and preferences stored in a personal device and in the network. These identities are categorized in three groups of identity, personal identity (PID), corporate identity (CID) and social identity (SID) based on the roles exercised by a person in real life. The PID can be used to identify ourselves in our very personal and commercial interactions, CID is used in our professional interactions, and SID in the social interactions. Fig. 3 shows the example applications of PID, CID and SID.

![Figure 3: Personal, Corporate and Social Identities](image)

Our approach suggest a de-centralized identity architecture, consisting of network components and the personal device of the user. Such an approach brings the user

---

3Liberty Alliance Project, http://www.projectliberty.org/
4Sxip Identity, http://www.sxip.org/
in the control of his services, allowing him to accept or deny access to privacy information. The mechanism builds on a personal user device, typically a mobile phone, providing the underlying infrastructure. A trusted and well-accepted third party will provide authentication and identity, thus become an identity provider (IDP). The identity provider will issue certificate to the user and allocates a secure identity space in the network, in addition to the sensitive PID space on the user device. An example of such an approach is the BankID partnership which provides a PKI supported identity for bank transactions on the SIM card of the mobile phone [21]. The parts of the user identities which need lower authentication requirements, for example social identities and preferences (SID) will be stored into the allocated secure identity space. To manage multiple credentials, a trusted third party/service provider can load additional identities to either the SIM card or a network identity space. Control of the sensitive data is given through the mobile network: In case of losing the SIM card, the card operator can disable the lost card and issue a new one, where identities can be reloaded through secure and signed SIM transactions.

With the identity subscription certificate users can access the network identity repository, e.g. service preferences located in the SID. Identities stored in this repository can give access to services (remote or proximity) that need medium or low level of security requirements. The main reason to store service and user preferences in the network is the availability of the network repository and the short response time, avoiding the costly and varying mobile/wireless link. Tab. 1 provides a summary of the identity types and their location. Personal identities (PID) are regarded as having high security, and thus will be stored in the personal device of the user, allowing him to control when and what PID information is released to service providers. Further guidelines supporting privacy handling are given in sect. 4.1.

The state or governmental organisations are traditionally the most accepted identity providers. With strong regulations in place, banks and mobile operators can also act as an identity provider (IDP). User should be able to make separate agreements with the selected IDP for the identity services. An IDP should maintain a strong trust relationship among its subscribers and with other IDPs. Such a strong trust relationship might be provided through the mobile network along with phone and SIM

<table>
<thead>
<tr>
<th>Identity</th>
<th>Example</th>
<th>Realisation</th>
<th>Location</th>
<th>Security Req.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>bank</td>
<td>certificate + key</td>
<td>SIM</td>
<td>high</td>
</tr>
<tr>
<td>PID</td>
<td>home admittance</td>
<td>home entry key</td>
<td>SIM</td>
<td>high</td>
</tr>
<tr>
<td>CID</td>
<td>visit admittance</td>
<td>temp. entry key</td>
<td>Network</td>
<td>medium</td>
</tr>
<tr>
<td>SID</td>
<td>preferences</td>
<td>foaf</td>
<td>Network</td>
<td>low</td>
</tr>
<tr>
<td>SID</td>
<td>attributes</td>
<td>foaf</td>
<td>Network</td>
<td>medium</td>
</tr>
</tbody>
</table>
card as the secure infrastructure for storing and exchanging identity information in the proposed solutions.

3.2 Data and service repository

This section will use the concept of role based identity presented in the previous section to enable service and document access for the scenario of fig. 1 and 2. We use the semantic description of social relationships to define service/document access rights as presented in fig. 4. Through the relation (here: mother of a child in class 2) Maria gets access to the photos taken by Frank, who is father of a child in class 2. The corresponding sequence diagram for this service access is given in fig. 5. Maria sends an authentication request to the service repository of class 2. The authentication request checks that she is related to Paul (father), and thus provides access to the class 2 repository and finally access to the photos taken by Frank. Authentication is key issue in this sequence. How to ensure that Maria is the mother of a child in class 2, and
that being allowed to access the photos Frank has taken? We suggest that Maria will use a "Web of Trust" mechanisms or an ID provider (see sect. 4) to proof that she is mother of Paul, and that these credentials can then be added to the "parents lists" of class 2.

Our service access example uses social relations (SID) of Maria, and corporate (here: school) relations (CID) of Paul. Service realisations based on social identities will use the semantic description of relationships, preferences and context information. Service access requiring PID information is subject to user involvement, as outlined in the next section.

4 Privacy and Authentication

This section will provide guidelines for privacy handling of user information in a semantically supported service environment. It will be up to the user to decide which of his information should be revealed to which group. Some parents would like to restrict the information like "who are my children" and "which school do they attend". Thus even the access to the foaf file would be restricted, and requires an authentication through Web of Trust or similar Internet mechanisms. Some users might require even stronger authentication based on key exchange, which is the focus of this section. The section conclude with a prototypical implementation of key exchange, based on an admittance scenario.

4.1 Privacy protection in a distributed architecture

A service related security infrastructure should just provide the information which is necessary to access the service, and should not compromise the privacy of the user. Our approach is based on two factors, (i) the authentication provisioning by an accepted identity provider and (ii) the distributed storage of personal, corporate and social identity information. Fig. 6 provides a sketch of the distributed approach, where an identity request is either answered from a formal ID provider or an ID-provisioning engine located in the service domain, in our example the home or corporate network. Keys and certificates of sensitive manner are stored in the personal device of the user. This will allow a strong authentication based on both a possession and a knowledge factor for e.g. bank applications. The reason to store the PID information in the mobile is to inform the user about transactions or access to other confidential data, and let him decide whether access should be granted.

Service preferences, user context and connectivity are stored in SIDs, allowing for adaptation of services. Privacy of those data should be ensured through mechanisms suggested by Sxip or Windows CardSpace, as introduced in sect. 3.

Following the distributed identity architecture of this paper, privacy is ensured through:

6Web of Trust RDF Ontology: http://xmlns.com/wot/0.1/
1. A definition of my social contacts based on foaf principles. Access to member information should be granted on a membership in a SID, e.g. the family, cycling friends or other interest communities. This will ensure that not all my memberships are public, but that e.g. family members have access to my family pictures.

2. The membership in these interest communities will also ensure an access to my preferences, which are of importance for just this community. Membership details will not be revealed to other SIDs.

3. Access to other preferences have to be granted on a case-by-case basis through mechanisms proposed by the Internet community (see sect. 3).

4. Sensitive information like payment will only be handled if accepted from the personal device of the user, either through profiles or specific user interaction. An example is payment by credit card, where the user might want to confirm that it’s okay to deduct 500 € from the card.

The following section will show a prototypical implementation of handling of sensitive information.

4.2 Prototypical implementation

We pointed out earlier that social identities (SID) and membership to social groups can be proven through Internet technologies e.g. Web of Trust. Here we will focus on stronger security requirements, involving personal identities. We take up an admittance scenario following the home admittance PID case of tab. 1. The prototypical implementation covers two aspects, (i) the use of a home entry key and (ii) the secure distribution of such a key through the mobile network. Steps in the prototypical realisation are illustrated in fig. 7 and are based on the following steps [22]:

![Figure 6: Authentication and privacy handling in a distributed manner](image)
1. Frank sends SMS to service number to allow Maria accessing his flat 15b. Example: SMS to 2034 "90838066 17.12.2006 1000-1400"

2. Service centre creates an service SMS installed on Maria’s phone 90838066: Admittance key transferred to NFC phone.

3. Service centre informs Maria: Access granted to flat 15b, My Road, My City on 17.12.2006 from 1000-1400h. Use phone to get entrance.

4. Maria enters flat 15b with help of the NFC phone. Frank might receive an acknowledge message when Maria enters the flat.

This example of an admittance control is prototyped using the functionality of Telenor’s PATS Innovation lab, Nokia’s 3320 phones with NFC shells and a simulated lock system. It comprises elements of seamless authentication, interworking between mobile and near-field communication and provides new and advanced services to the end user. Frank is authenticated in a seamless matter when sending the SMS, interworking is provided through PATS’ lab based install messages. The users, Maria and Frank, experience admittance based on SMS exchange as an advanced new service.

The service scenario implemented above uses the mobile operator as identity provider. SMS is treated as a secure matter for distribution of admittance certificates. While this role of a mobile operator is accepted in some regions, it can’t be generalised. This paper suggests that an identity provider should take the role of providing certificates, and that the SIM card is managed by the identity provider.

The current implementation demonstrates the distribution of PIDs, thus ensuring the privacy of the most sensitive parts of the user identity. Implementations of SID and CID are subject to further work, and will require a more detailed analysis of potential security threats.

http://www.pats.no
5 Conclusions

Ubiquitous computing and the pervasive Internet have enabled service access in every situation. However, adaptation to the user needs is purely handled, and service specific security implementations are only found for specific services. The paper introduces an integrated mechanism for identity provision facilitating both remote and proximity service access. A semantic description of user preferences, context and connectivity is used to describe the social identity of the user to be stored in a secure user identity space in the network. Sensitive information, defined as personal identity, is suggested to be stored in the SIM card of the user’s personal device, providing him with full control over the usage of such service.

Multiple factors of authentication mechanisms are employed to address different levels of security requirements. The paper also demonstrates service access architectures using the proposed identity mechanisms, both for proximity and remote services. As such it combines the I-centric and service centric service world. The prototypical implementation of SMS-based admittance key distribution covers two aspects, (i) the use of a contactless home entry key and (ii) the secure distribution of such a key through the mobile network.

References


INFORMATION MODELLING AND SHARING FOR RFID-EQUIPPED OBJECTS

Harri Hämäläinen
Lappeenranta University of Technology
Faculty of Technology Management, Laboratory of Communications Engineering
P.O. Box 20
FIN-53851 Lappeenranta
Finland
harri.hamalainen@lut.fi

ABSTRACT
This paper shortly describes the hybrid-model for RFID applications. This model enables passive tags to exchange information between themselves and decisions related to e.g. handling and warehousing can be made based on product information and created rules. This information can be stored either in the tag memory or in distributed servers where the information is linked to each object based on the unique IDs. This model requires some additional functionality and semantics to RFID middleware.

KEYWORDS
RFID, database, middleware, hybrid-model

1. INTRODUCTION

Smart environments, containing wireless sensor networks, are said to be one of the next steps in systems automation in various environments. RFID has also been a hot topic in information technology for a while and it is expected to be widely used in logistics. However, RFID identification is not yet that common on item level tagging, but larger objects such as pallets are already being identified.

To gain the best possible advantages from RFID, whole supply chain and all of its trading partners have to be considered. Information that concerns only the life cycle of a single product does not bring out all the benefits of the technology. More interesting approach is the hierarchical information combined with sharing it with the trading partners. In this case products and objects can be linked to each other which enables the users to get more specific and valuable information about the products. Each product may have links to the raw material they were produced from and during the transportations they belong to a larger unit. All the information that is provided by this unit can be shared among the products that are a part of it.

Storing information for each item creates a huge amount of information. Therefore, there has to be a functional solution for linking and sharing the information between different operators. This raises a question about who is responsible for storing and controlling the access to the information. One of the challenges will also be the trust for the system as well as the information and its validity.
2. RFID USAGE MODELS

The RFID usage models refer to the information architecture as well as how and where the information of a product is stored. These models make it possible for different sizes of companies to utilize the RFID technology for various purposes and applications. In some cases you may just need to write the information in a tag and read it from time to time whereas covering the whole supply chain requires a lot more complicated systems where information is distributed and linked between numerous information services.

The simplest solution is a tag-based solution where the information is stored in the tag itself. In this case there is no need for centralized systems, but each of the parties must still be able to understand the format of information. The content that is used for storing the parameters could be e.g. XML-format or some other, more lightweight structure. This model however requires that tags have rewritable memory and causes that the information stored in the tag is only available when the tag is within the read-range.

To take advantage of the possibilities of the technology, information sharing between trading partners plays a significant role. The amount of information that is linked into a single object is often so large, that it cannot be stored in a rather limited memory of a tag. A non-profit organization called EPCglobal provides industry standards for this type of network architecture [EPC05]. Objects are identified on item level using a unique code that is constructed from the IDs of EPC manager, object class and serial number of the item. Object Name Servers (ONS) direct the queries of objects to appropriate information servers that store the actual information of the product and its lifecycle.

Our idea of utilizing RFID is based on hierarchical modeling of product information. While the tag-based solution requires that the information must be stored in a single tag, including the information of the raw materials and other related events, the EPC-model gives a chance for linking information to each other. This can mean different things on different phases of the life cycle of the product. For us this hybrid model means highly dynamic approach where the information stored in a tag and in the servers can be combined and updated easily. In our approach there can also be seen some of the advances of sensor networks. While speaking of RFID technology, the usage of this kind of approach would traditionally require active tags that have batteries to be able to send information to each others. Our approach in the research is to share the functionalities of operational tags to the others by taking advantage of RFID-middleware that can act as an intermediate that passes the information from tag to another if needed and an additional component that takes care that conditions that are set are filled. In this model the information stored either on tag or server as well as rules for information handling plays a significant role.

REFERENCES

SERVICE BASED TASK MIGRATION IN UBIQUITOUS ENVIRONMENT

Jari Porras
Lappeenranta University of Technology
P.O.Box 20, 53851 Lappeenranta, Finland
Jari.Porras@lut.fi

ABSTRACT
In this short concept paper an approach for utilizing distributed resources for remote execution of tasks in ubiquitous environment is presented. The goal for this approach is to define computational services on network devices so that they can be utilized by mobile devices within their vicinity.

KEYWORDS
Task migration, remote execution, ubiquitous environment

1. INTRODUCTION

Ubiquitous computing is currently one of the main trends in the world of communications. The environment around us is becoming more and more ambient. Computing technologies recede into the background like Mark Weiser envisioned in his ground breaking article of the computer of the 21st century. Important part of this ubiquitous world is the interface to the environment. Mobile devices like smart-phones, PDAs or laptops are considered suitable for this task because of their ability to move in the ubiquitous environment. The high penetration of mobile phones as well as their usability and personal nature make them the most viable interface to the ubiquitous world.

However, the use of ubiquitous environment is restricted by many aspects. Slow processors at the mobile phones as well as range and bandwidth limited communication technologies make the exploitation of the ubiquitous environment challenging. Mobile devices are also energy limited due to their batteries. As communication and computation both require energy the energy consumption set the limits for the usability of mobile devices in ubiquitous environment. Energy needed for communication can be reduced by the use of local communications as well as optimized communication patterns. Energy consumed for computations is more challenging as the intelligent decisions in ubiquitous environment require more and more computations. These computations can be performed locally at the mobile device or remotely at some ubiquitous computing resource. Local computations are limited by the slow processor at the mobile device whereas remote execution suffers from overheads of the task migration. In this paper we consider software solution for the task migration in the ubiquitous environment. We define the parts of the system as well as their operation.
2. SERVICE BASED TASK MIGRATION

MobiGRID project aims in combining ubiquitous environment with remote execution. We divide the remote execution into three types, namely operation migration, service based task execution and computational service. Operation migration is specific to our PeerHood platform and means that part of the PeerHood operations can be migrated to another PeerHood device. Service based task execution is presented in Figure 1 and is the first phase of the MobiGRID project. In our example camera is used for creating some task for the ubiquitous environment. Mobile phone uses PeerHood connectivity solution to discover devices and services in its neighborhood. Mobile device either discovers required services or it inquiries those from the network. Services need to have a service description and a suitable service interface. Although our task migration approach is first demonstrated with image based problems both task and service descriptions are defined in a general manner. Computational service means that we would have general computational services where we could submit tasks to be executed.

Task migration has been proposed by many researchers before. Gitzenis and Bambos proposed in their paper [1] a task migration from the energy consumption point of view. Several parameters concerning remote execution were considered and simulations were run with different setups. Similar approach is presented by Rudenko et. al in [2]. A real task migration experiment is presented in [3] where the authors migrate an image analysis problem from mobile phone to a local computing resource. Remote execution is largely and more generally addressed in GRID computing field.

REFERENCES

Call for Papers

6th Workshop on Applications of Wireless Communications (WAWC'08),
August 14, 2008 - Lappeenranta, Finland

in conjunction with the 17th Summer School on Telecommunications
at the Lappeenranta University of Technology, Finland

Important dates
Abstract submission deadline: January 14, 2007
Paper submission deadline: February 8, 2008
Notification of acceptance: March 31, 2008
Revised papers due: May 5, 2007

Overview
The research related to wireless communication and its use, concentrates on providing tools for application developers to create new and better applications and services e.g. a better way to handle connection forming, a new way to manage personal information or a new approach to model context information. There is less emphasis on the actual applications that use these new methods i.e. applications that benefit from context-awareness, use the new communication model, adapts to each person individually. WAWC'08 concentrates on such applications and services. The major emphasis of the workshop this year is security of wireless applications. What are the real security threats towards wireless services? Do the existing research results provide feasible solutions to these threats?

WAWC'08 is a one-day workshop featuring invited presentations, refereed paper presentations, student work presentations and a work in progress discussion. WAWC'08 will be held in conjunction with the 17th Summer School on Telecommunications that concentrates on various issues of the Wireless World. The Summer School program also includes a 24h programming event, Code Camp. Participants of WAWC'08 are welcomed to join also in Summer School events.

Scope/Topics
Papers about security in applications and services using wireless communication should concentrate at one of the following topics.

- Authentication
- Privacy threats and protection
- Trust
- Automation and Simplicity
- Lightweight cryptography
- Availability
- Threat analysis
- Middleware
- Seamless connectivity

- User-Centric security
- Feasible security solution
- Security Management
- Ambient Intelligence
- Pervasive/Ubiquitous Communications
- Wireless sensors security
- BAN/PAN/LAN
- Touch zone security
- Payment systems

Please feel free to contact the program chair at WAWC@lut.fi to determine the appropriateness of the topic of your paper.
What to submit

Abstract submission:
Abstract should be maximum of one page long describing the main idea of the paper. The abstract submission is used as a pre-filtering method to guarantee that the reviewers will get only papers suitable for the workshop.

Paper submission:
The submission can be either full paper (maximum of 10 single-spaced A4 pages, including figures, tables and references using point 12 font), including figures, tables and references using point 12 font). Details of the paper format can be found on the workshop website.

A good paper demonstrates that:

- Application area presented is or will be of greater interest of majority
- Wireless communication has a significant role in the solution
- The realization of the security solution presents a new way of doing things

WAWC'08 among most conferences and journals requires that papers must be original and published only in this conference. Proceedings will be published in the Acta Universitatis Lappeenrantaensis series of Lappeenranta University of Technology.

All selected papers must be paid in advance and presented in the conference.

How to submit

All submissions for WAWC'08 will be electronic, in PDF format. Please, submit papers through the paper submission system at the workshop website. Authors will be notified of the receipt of submission via email. If you do not receive notification, contact WAWC@lut.fi.

Review process

All the papers will be reviewed by at least three members of program committee.

Registration

Complete program and registration information will be available in May 2008 on the Workshop Web site http://www.it.lut.fi/WAWC. The information will be in both html and a printable PDF file formats.

Questions

Inquiries concerning the workshop to WAWC@lut.fi.

Organized by the Laboratory of Communications Engineering, Lappeenranta University of Technology.
ACTA UNIVERSITATIS LAPPEENRANTAENSIS


228. HURMELINNA-LAUHKANEN, PIA. Dynamics of appropriability – finding a balance between efficiency and strength in the appropriability regime. 2005. Diss.


233. BJÖRK, TIMO. Ductility and ultimate strength of cold-formed rectangular hollow section joints at subzero temperatures. 2005. Diss.


237. RITVANEN, JOUNI. Experimental insights into deformation dynamics and intermittency in rapid granular shear flows. 2006. Diss.


239. ELFVENGREN, KALLE. Group support system for managing the front end of innovation: case applications in business-to-business enterprises. 2006. Diss.

240. IKONEN, LEENA. Distance transforms on gray-level surfaces. 2006. Diss.


244. AARNIO, TEIJA. Challenges in packaging waste management: A case study in the fast food industry. 2006. Diss.

245. PANAPANAAN, VIRGILIO M. Exploration of the social dimension of corporate responsibility in a welfare state. 2006. Diss.


247. SALMELA, NINA. Washing and dewatering of different starches in pressure filters. 2006. Diss.


254. VEHVILÄINEN, JUHA. Procurement in project implementation. 2006. Diss.


256. RAUMA, KIMMO. FPGA-based control design for power electronic applications. 2006. Diss.

257. HIRVONEN, MARKUS. On the analysis and control of a linear synchronous servomotor with a flexible load. 2006. Diss.


259. LAITINEN, RISTO. Development of LC-MS and extraction methods for the analyses of AKD, ASA, and rosin sizes in paper products. 2006. Diss.


262. SOININEN, AURA. Patents in the information and communications technology sector – development trends, problem areas and pressures for change. 2007. Diss.

263. MATTILA, MERITA. Value processing in organizations – individual perceptions in three case companies. 2007. Diss.

264. VARTIAINEN, JARKKO. Measuring irregularities and surface defects from printed patterns. 2007. Diss.


266. SEKKI, ANTTI. Successful new venturing process implemented by the founding entrepreneur: A case of Finnish sawmill industry. 2007. Diss.


268. BUTYLINA, SVETLANA. Effect of physico-chemical conditions and operating parameters on flux and retention of different components in ultrafiltration and nanofiltration fractionation of sweet whey. 2007. Diss.