Nor Shahniza Kamal Bashah

Mobile Service Architecture in Future Mobile Environments

Thesis for the degree of Philosophiae Doctor

Trondheim, February 2013

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

ISBN 978-82-471-4170-0 (printed ver.)
ISBN 978-82-471-4172-4 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2013:39

Printed by NTNU-trykk

Especially for my late father,

Kamal Bashah bin Abd Razak

(11^{th} January 1941 – 19^{th} February 2010)

Al-Fatihah

Abstract

Nowadays, most mobile phones are equipped with multiple wireless access technologies, which allow the user to maintain connections while moving across networks (e.g. having a telephony service in GSM while moving to a WLAN area). The IEEE 802.21 Media Independent Handover (MIH) Services [1] is a standard aiming at facilitating handover between these heterogeneous networks. In the current situation, in order for the mobile device to detect other wireless access network (e.g. WLAN), the user has to keep the WLAN interface continuously on and this will cause battery drain. With IEEE 802.21, the mobile device will have continuous access to the MIH function via a Service Access Point (SAP) on the GSM network and can hence receive information about the availability of the WLAN hot spot. The WLAN interface can then be turned on and the connection can be established.

Unfortunately, the standard only provides the information for network connectivity but is incapable of supporting service continuity. More specifically, the standard does not enable the user to seamlessly change from GSM telephony to IP telephony when entering a WLAN hotspot.

To remedy the situation this thesis has analysed future mobile scenarios and proposes a *mobile service architecture*, which enhances the availability of services at the same time as service continuity is ensured. The necessary functional entities both on the network and terminal side are identified and described. The thesis has also conducted thorough analysis and realisation of the two main elements of the mobile service architecture, namely *Service Continuity* and *Service Discovery*.

To realise Service Continuity it is proposed to extend the functionality of the IEEE Media Independent Handover (MIH). More specifically, it is proposed to extend the information database with service information and to include service information in the information exchange between the mobile client and the MIH.

Regarding Service Discovery it is an innovative discovery scheme capable of handling the following characteristics:

- Similar services with different names in different languages
- Different services with similar names
- Partially similar services
- No restriction regarding the name length
- Anybody can introduce anything as a service anytime

To verify the validation of the three proposed items:

- The future mobile service architecture
- The Service Continuity
- The Service Discovery

partial but sufficient implementations have been carried out. Consequently, it is possible to conclude that the three proposed items are both sound and feasible.

Preface and Acknowledgements

This thesis is submitted in partial fulfillment of the requirements for the degree Philosophiae Doctor (PhD) at the Department of Telematics, Norwegian University of Science and Technology (NTNU). This PhD is fully funded by the Ministry of Higher Education (MOHE) and Universiti Teknologi MARA, Malaysia. The work was carried out at the Department of Telematics during the period August 2007-December 2010. My main supervisor is Prof. Dr. Do van Thanh (NTNU and Telenor R&D), and my co-supervisor is Prof. Dr. Steinar Hidle Andresen (NTNU).

My deepest gratitude is particularly addressed to Prof. Dr. Do van Thanh for his great assistance, encouragement and invaluable support during the three years period of my study. His guidance and comments of my work have always been a motivation for me especially in delivering quality of work and writing. I am also grateful to my co-supervisor Prof. Dr. Steinar Hidle Andresen for his good advice and interesting discussions especially in Network field. Many thanks also to Ivar Jørstad (Ubisafe), Natalia Kryvinska (University of Vienna, Austria), Tore Jønvik (HiO) and Do van Thuan (Linus) for the good input during the collaboration of writing papers. I would also like to thank the administrative and technical team of this Department especially Randi Flønes, Mona Nordaune & Pål Sæther and both Head of the Department, Norvald Stol (2007-2009) & Poul E. Heegaard (2009-2010) who had always help me from the first day I joined Telematics. Thanks also to all my PhD colleagues, Linda, Martin, Jing, Patcharee, Màtè and Razib for making my work life in Telematics more interesting. Special thanks also to my Master students, Atif and Imran for the good deliveries in your Master thesis work.

Finally, thanks to my family in Malaysia especially my Mother Siti Hajar binti Md. Rejab for their continuously support and prayers throughout the years. Last but not least, my sincere thanks to my dearest husband, Rizal for his patience and continuous encouragement during this PhD journey and to our four little angels Raziq, Rizqin, Rihana and Raihan, for all the joy and happiness you shared in my life.

Trondheim, December 2010

Nor Shahniza Kamal Bashah

List of Publications

The papers comprising the main parts of this thesis are found in Part II.

Main papers

Paper 1 Kamal Bashah, N.S, Jørstad, I. & van Do, T (2009). “Enabling Service Continuity on Future Mobile Services”, 4th International Symposium on Wireless Pervasive Computing (ISWPC2009), Melbourne, Australia, February 11-13 2009, ISBN 978-1-4244-2965-3

Paper 2 Kamal Bashah, N.S, Jørstad, I. & van Do, T (2009). “Service Naming in Future Mobile Environments”, 38th International Conference on Parallel Processing (ICPP2009), Vienna, Austria, September 22-25 2009, ISBN 978-0-7695-3803-7

Paper 3 Kamal Bashah, N.S, Jørstad, I. & van Do, T (2010). ”Service Discovery in Future Open Mobile Environments”, 4th International Conference on Digital Society (ICDS2010), St.Maarten, Antilles, Netherlands, 8-15th February 2010, ISBN 978-0-7695-3953-9

Paper 4 Kamal Bashah, N.S, Jørstad, I. & van Do, T (2010). “An Open and Extensible Service Discovery for Ubiquitous Communications Systems”, Networked Services and Applications – Engineering, Control and Management (EUNICE 2010), Trondheim, Norway, 28 – 30th June 2010, LNCS 6164, pp. 272-273, 2010, ISBN-13 978-3-642-13970-3

Paper 5 Kamal Bashah, N.S, Bhatti, A, Choudhary, I.A, Jørstad, I. & van Do, T (2010). “Service Discovery for Mobile Multi-domain Multi-language Environments”, The 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010), Niagara Falls, Canada, 11 – 13th October 2010, ISBN 978-1-4244-7741-8

Paper 6 Kamal Bashah, N.S, Kryvinska, N & van Do, T (2010). “Service Discovery in Ubiquitous Mobile Computing Environment”, The 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2010), Paris, France, 8 – 10th November 2010, ISBN 978-1-4503-0440-5

List of Publications

Paper 7 Kamal Bashah, N.S, Jørstad, I, van Do, T, Jønvik, T & Van Do, T (2011). ”A Mobile Service Architecture for improving Availability and Continuity”, The 2011 IEEE Symposium on Computers & Informatics (ISCI2011), Kuala Lumpur, Malaysia, 20^{th} – 22^{nd} March 2011, ISBN 978-1-61284-690-3

Paper 8 Kamal Bashah, N.S, Kryvinska, N & van Do, T (2011). ”Service Continuity across heterogenous networks”, The International Conference on Software and Information Engineering (ICSIE 2011), Kuala Lumpur, Malaysia, 17^{th} – 19^{th} June 2011, ISBN 978-0-7918-5973-5

Paper 9 Kamal Bashah, N.S, Kryvinska, N & van Do, T (2012). ”Novel Service Discovery Techniques for Open Mobile Environments and their Business Applications”, Third International Conference on Exploring Service Science (IESS 1.2), Geneva, Switzerland, 15^{th} – 17^{th} February 2012, Springer, Lecture Notes in Business Information Processing (LNBIP 103), pp. 186–200, 2012, ISBN 978-3-642-28226-3

Paper 10 Kamal Bashah, N.S, Kryvinska, N & van Do, T (2012). ”Quality Driven Service Discovery Techniques for Open Mobile Environments and their Business Applications”, Journal of Service Science Research, ISSN 2093-0720 (print version), ISSN 2093-0739 (electronic version), Journal No. 12927, *accepted*

Contents

Contents

List of Figures

List of Figures

List of Tables

List of Tables

Glossary

Name-to-object association – "... is called a name binding." [2]

Name resolution – "... the process to determine the object associated with a name in a given context."

Namespace – "An abstract container providing context for the items (names, or technical terms, or words) it holds and allowing disambiguation of items having the same name (residing in different namespaces)."

Naming convention – "A convention for naming things."

Naming policy – "... provides guidelines aimed at ensuring the appropriate naming."

Network service – "... is a service offered to the user by a network system."

Nomenclature – "A set or system of names or terms including the rules used for forming the names as those used in a particular science or art, by an individual or community, etc."

Semantic matching – "... requires a service description complete with the ontology definition and a service engine with reasoning mechanism." [3]

Service – "... is a mechanism enabling the end-user's access to one or more capabilities." [4]

Service availability – "... the time when the service can be accessed."

Glossary

Service classification – “A process of assigning a class to a service, indicating the domain/category a service belongs to.”

Service continuity – “... the ability for a user to maintain an ongoing service during mobility across domains, networks and devices.”

Service discovery – “A process of obtaining a set of services which can possibly fulfill a user request.” [5]

Service matching – “The process of comparing the service request against the available service advertisements and determining which service best satisfies the request.” [6]

Service naming – “A method on how the service is called.”

Service naming scheme – “A system for assigning names to services.”

Syntactic matching – “... will only do the comparison based on the attributes or interfaces...” [3]

Part I – Main thesis

Part I – Main thesis

1. Introduction

1.1 Motivation

For the past three decades since the first mobile telecommunication service was introduced in the early 80's we have seen tremendous advances in mobile services which would not be possible without progress in wireless technologies. Mobile phones have evolved from being a terminal offering basic telephony service to the user to become an intelligent device offering not only telecommunication but also other fancy and advanced services such as navigation, music player and much more.

Most of mobile phones are nowadays equipped with multiple wireless access technologies, e.g. GSM, GPRS, UMTS, WLAN, Bluetooth, etc. which enable them to be connected to multiple networks at the same time. In principle, this should allow users to access to a wider range of services. However, the potential of the enhanced connectivity is not fully exploited and many improvements can be done to improve the availability and the continuity of the services and to promote the user's freedom of choice of services based on different factors such as performance, price, security, role, etc.

To clarify the point let us consider the state-of the-art mobile network system as depicted in Figure 1. Through the circuit-switched CS domain of the mobile network consisting of the traditional GSM network the user can access and consume the voice and SMS services. Indeed these two services are both the first and most dominant mobile services. With the introduction of the packet-switched PS domain consisting of GPRS [7], 3G [8], HSPA [9] the user can have access to the Internet and enjoy a wide range of services like information, commerce, social communities, etc.

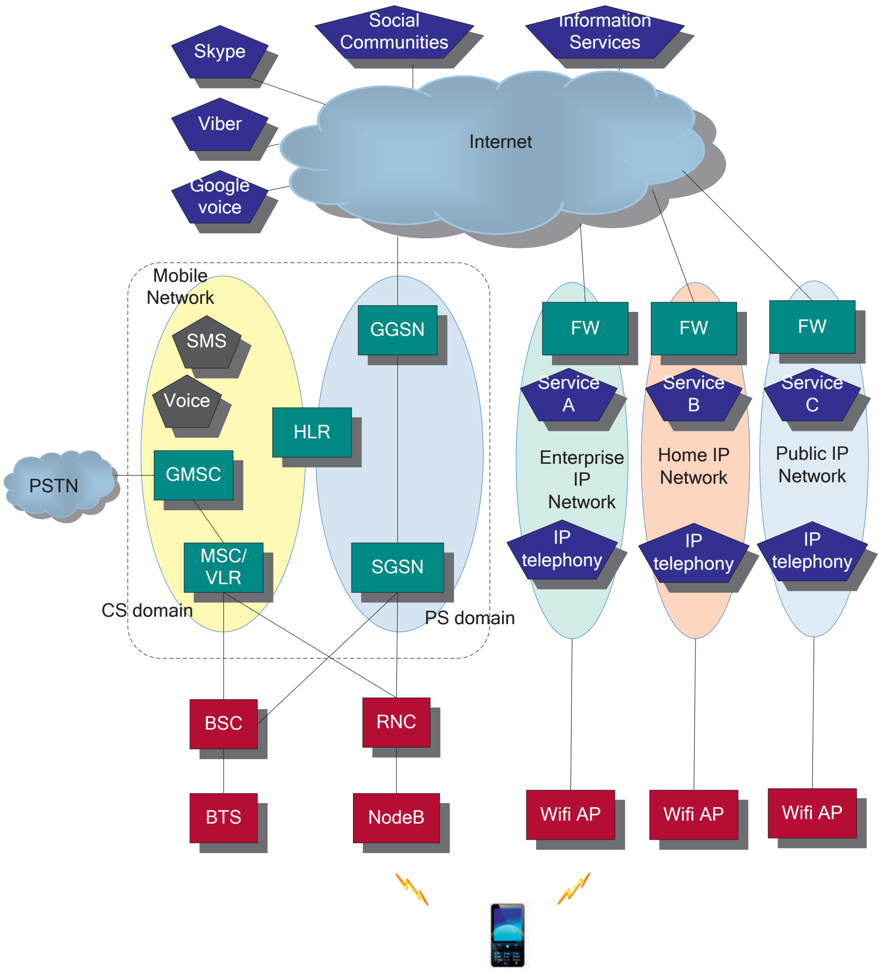

Figure 1 Today's user access to mobile services

Newer mobile phones are quite often equipped with wireless LAN 802.11 [10] also called WLAN or WiFi enabling the connection to the Internet via the enterprise IP network or the home IP network or any authorized IP network. The user can hence have access to Internet services for free or for lower fee.

Introduction

Recently, as shown in Figure 1, a plethora of VoIP applications such as Skype, Viber, mofree, Fonea, etc. have emerged in the Internet and offered free or cheaper phone calls via 3G and WLAN. Some of them do also offer SMS. These VoIP applications are client software that can be easily downloaded and installed on the mobile phones by the user. They can either be automatically switched on at power on or initiated by the user. They are interacting with a centralized or distributed server part located in the Internet to offer voice communication.

Although there is a big variation in terms of sound quality, user friendliness these VoIP applications are all dependent on the quality of the 3G or WLAN connection resulting generally to lower quality than the GSM voice service. The main drawback is the lack of integration with the GSM telephony service and the user does not make or receive calls using their mobile phone number but a URI (Uniform Resource Identifier) such as bob@ntnu.no.

These VoIP applications reside on the Internet, which can be viewed as an independent and ubiquitous service layer available over all access network technologies. These applications on the Internet are often referred to as Over The Top (OTT) applications and service continuity is preserved as long as there is continuous network connectivity. This thesis does not focus on these Internet services but on services residing in the mobile, enterprise and home networks.

In enterprise and home IP networks IP telephony service is also getting more and more popular due to its low flat rate and its improving quality. Both network operators and ADSL operators do offer their IP telephony service. In addition there are also autonomous service providers such as Telio, Ventelo, Phonzo, IPonNet, etc. that offer only IP telephony service either on PCs or wireless handsets as shown in Figure 1. These telephony services are currently separate from the GSM voice service even in the case where the operator is the owner of the mobile network and the broadband ADSL network. To enjoy cheaper calls the users have to handle two incompatible telephony services.

Seen from the user it would be nice if the disparate telephony services i.e. GSM and IP telephony services are unified and service continuity is provided between them. With service availability the most appropriate telephony service for each situation will automatically be selected for the user. With service continuity a user with an ongoing GSM call on the mobile phone will be automatically switched from GSM to the enterprise or home network when entering the WLAN coverage area. The ongoing GSM call will be smoothly replaced by a VoIP call without disrupting the user's conversation. The user's motivation for carrying such a call change is the low flat rate[1] of the IP telephony service at enterprise or at home. By moving mobile calls to the broadband fixed network the user may save money and improve the family budget. Another advantage is the reuse of the mobile phone at home, which removes the need for additional cordless telephone handset.

For the mobile operator, a call transfer from GSM to VoIP may constitute a way of offloading the heavily loaded mobile network during peak hours. Furthermore, the operator will be in a better position to offer bundled service packages including both mobile and fixed subscription which increase customer's loyalty and reduce churn.

For the mobile user it might also be relevant to discover and use all services offered by the visited network. Some services such as voice, messaging, chat, etc. are familiar to the user and are chosen due to several factors like price, performance, security, etc. Other services are more specific to the visited network such as local information, local offers, IPTV, games, etc. and might be quite interesting for the visiting mobile user.

[1] A flat fee, also referred to as a flat rate or a linear rate, refers to a pricing structure that charges a single fixed fee for a service, regardless of usage

Moreover, users may visit different places such as family, relatives, friends, public places, etc. and get granted access to different IP networks offering various services which can be identical, equivalent or different with the ones they are used to. It might hence be quite interesting for the users that these services are discovered and made ready for usage by the users.

To summarize, the following scenario would be highly desirable for mobile users:

- As the user moves and changes location the mobile terminal will detect all available heterogeneous access networks not only mobile networks standardized by 3rd Generation Partnership Project (3GPP) like GSM, GPRS, 3G, HSPA, LTE [11], etc. but also enterprise, home or other networks using wireless LAN family, 802.16 wireless Broadband series commercialized as WiMAX, 802.20 Mobile Broadband Wireless Access (MBWA).
- The mobile terminal will thereafter select and connect to one or more networks offering the services that match the user's needs and preferences before terminating the connections with the former networks. The user's preferences could be price, performance, security, etc.
- In the case where there are ongoing services, e.g. voice, SMS, mail, banking, commerce, etc. service continuity will prevail such that the user can continue using the services without disruption.
- In the case where no service is in use, appropriate preparations will be carried out such that the user's preferred services which are available in the visiting networks are made ready for usage.
- If the user wants, the specific services offered by the visited network shall also be presented to the user in a comprehensive way.

Although the described scenario may sound familiar and trivial there are still some challenging issues that need to be resolved as follows:

- Seamless handover between access technologies
- Seamless call transfer between two telephony services
- Access to service information
- Service discovery for open environments
- Unifying mobile service architecture

The mentioned issues will now be successively studied and the ones considered in this thesis will be clearly indicated.

Seamless handover between access technologies

Although the number of accessible wireless networks keeps increasing it does not automatically mean that the user can benefit them. Indeed, the user's mobile terminal must first discover the available networks before using them. For that, all the mobile phone's radio interfaces, e.g. WLAN, WiMAX, Bluetooth, etc. have to be turned on constantly and keep draining the battery. In fact, in order to save battery, most users do turn off the unused radio interfaces and keep only the operating one. Consequently the mobile phone is prevented from discovery wireless networks and services that might be relevant to the user.

A few solutions to this problem have been proposed but unfortunately, they are not quite satisfactory. For example, the mobile phone can be programmed to switch on WLAN when the signal quality of GSM has deteriorated under a certain level. Unfortunately, this solution does not enable the detection of WLAN networks when the GSM signal is still good, which is most of the time due to the current great coverage of GSM nowadays. The user might want to change to WLAN due to performance or price and not due to signal quality or coverage of GSM. Another solution is to put the WLAN interface in sleep mode and wake it up periodically. This solution helps obviously to reduce the power waste but is far from being optimal. Moreover, and increasing number of radio interfaces will increase the power consumption and this solution will be unacceptable.

To address this problem the Institute of Electrical and Electronics Engineers (IEEE) has introduced the 802.21 Media Independent Handover (MIH) Services [1]. The MIH is a standard aiming at facilitating handover between these heterogeneous networks. With IEEE 802.21, the mobile device will have continuous access to the MIH function via a Service Access Point (SAP) on the GSM network and can hence receive information about the availability of the WLAN hot spots. The WLAN interface can then be turned on and the connection can be established.

Conclusion: *The seamless handover between access technologies is covered by the standard IEEE 802.21 MIH and will not be considered in this thesis.*

Call transfer between two telephony services

With the 802.21 MIH standard service continuity will be preserved for Over The Top applications in which the service client and the service server part remain unchanged and only the access networks are shifted. For example, when the mobile terminal is moving from a 3G network to a WLAN network the MIH will provide sufficient information to carry out seamless handover between 3G and WLAN and hence ensure that an ongoing Skype call can continue without disruption.

However, if the user has an ongoing GSM voice call and wants to shift to IP telephony when entering the WLAN IP network the information provided by the 802.21 MIH is not sufficient to ensure service shift and continuity. Indeed, the 802.21 MIH ensures only continuous connectivity between 3G and WLAN. In order to carry smooth and seamless service shift between GSM voice and IP telephony the IP telephony service in the IP network has to be discovered and the IP client has to be initiated and prepared for call establishment. A new call leg will then be established with the IP client before the termination of the former GSM call leg.

The call transfer between GSM and IP can be realized by introducing IMS (IP Multimedia Subsystem) as shown in Figure 2. The Mobicome project has implemented and demonstrated a call transfer between a GSM call on a mobile phone and an IP telephony client on a PC [12]. A media server realized by the Snowshore Server, now called Dialogic IP Media Server [13] is used to create and bridge multiple RTP media streams and hence enable the establishment of a new calling leg without disrupting the call. Although not implemented due to limitations of the mobile phones, call transfer between GSM voice service and an IP telephony client on the same phone is in principle the same as on two separate devices and should be possible.

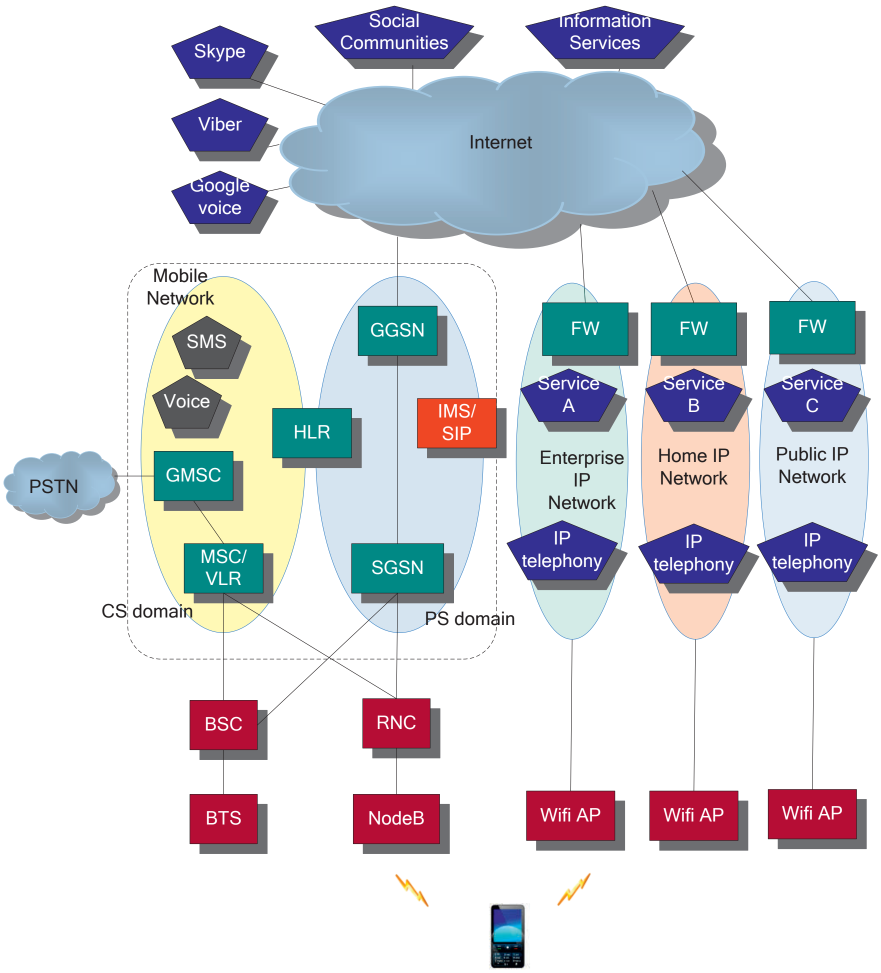

Figure 2 Using IMS to unify voice call service

Conclusion: *The call transfer between two telephony services was already demonstrated by the Mobicome project and will not be considered in this thesis.*

Access to service information

The challenge now is to find out which telephony services are available in order to initiate and use them. In the case of the enterprise or home network, the IP of the available telephony service is known and the mobile phone can be configured to initiate the IP telephony client upon handover from 3G to enterprise or home network. However, if the goal is to provide service transfer between any networks then the mobile phone must somehow have access to information about the services available on the visiting network. This is not possible today and new solutions are required.

Conclusion: *The access to service information necessary for service availability and continuity is currently not provided to the mobile terminals and constitutes one of the main issues addressed in this thesis.*

Service discovery for open environments

Let us assume that the service information is somehow published and the mobile phone can easily access them. This service information will, however be useless to the mobile terminal unless both the syntax and semantics of the service information is known by the mobile terminal. Indeed, at least the format and convention of the service naming and typing must known by the mobile terminal if it wants to know which services are available. For example in UPnP (Universal Plug and Play) the `CallManagement` service [14] is a UPnP service that allows control points to use the telephony features (e.g. voice call, video call, and data transfer, etc.). The service type for `CallManagement` is defined as follows:

```
urn:schemas-upnp-org:service:CallManagement:1
```

The UPnP standard service types are denoted by `urn:schemas-upnp-org:service:` followed by a unique name assigned by a UPnP forum working committee, colon, and an integer version number. To discover a UPnP service a string matching is carried out on the service type. In GSM, the service code for telephony and Emergency calls are respectively 11 and 12 [15] and service discovery is done by comparing the service codes.

Generally, most of state-of-the art service discovery system such as UPnP, Jini, dlna, UDDI, Web Service Dynamic Discovery, etc. are based on service type or name string matching on standardized well defined service types. The service type or name is usually uniquely defined and standardised by a centralized body such as UPnP forum, OASIS UDDI technical committee, etc. and all the service providers have to comply with it.

In an open deregulated heterogeneous network environment where a service can be anything and even the end-users can be service providers, current centralized and standardized service discovery systems are not quite appropriate. Indeed, it should be possible for anyone to offer any service with any name. A more open service discovery system allowing dynamic introduction of services without demanding approval from a centralized body is requested.

Conclusion: *In an open heterogeneous network environment where anyone can be a service provider and a service can be anything with any name not every service can be discovered by existing service discovery systems. This unsolved issue will be studied in this thesis and innovative solutions will be proposed.*

Unifying mobile service architecture

As mentioned earlier the users are now facing multiple competing services by different service providers on different access networks. For example, the telephony service may be offered by different service providers using different implementations with different components and different arbitrary names e.g. Telio, Phonzo, IPonNet, etc. They are perceived as equivalent by the users since they all offer voice communication. However, there is no standardization body which formally approves their equivalence. In fact even the service providers tend to differentiate and advertise the uniqueness of their services.

From the user point of view, it would be nice if they can self define a generic service e.g. telephony, messaging, chatting, etc. which unifies all the disparate but equivalent services.

To realize this vision it is necessary to have a mobile service architecture and framework incorporating and unifying existing equivalent services, which does not exist today.

Conclusion: *There is no mobile service architecture and framework which allows the user to unify the disparate equivalent services. This thesis is focusing on this unresolved issue and an appropriate mobile service architecture and framework will be proposed.*

1.2 Problem statements and goals

From the described scenario the problem statements can be extracted and summarized as follows:

- There is no adequate architecture and framework for unifying mobile services, which allows the users to define equivalent services and thereafter use them in a combined way.
- There is currently no adequate solution that provides service continuity across multiple heterogeneous networks offering disparate equivalent services.
- The support for handover provides by the IEEE 802.21 Media Independent Handover (MIH) is not sufficient to ensure service continuity when service transfer is requested.
- To be able to transfer services the mobile device needs to have access to service information for a location, which is not present today.
- There is no adequate service discovery that can handle new and unpredictable services with arbitrary names introduced and provided by different service providers.

The goals of the thesis are as follows:

- To contribute to the design and modelling of the architecture of future mobile services.

- To study, design and implement capabilities and functions that support service continuity in a multiple network system environment.
- To study, design and implement a service discovery for future mobile services.

1.3 Methodologies

1.3.1. Research methodology

The research methodology used in this thesis is based on the design science research process [16]. The design science is inherently a problem solving process. Hence, it is important to define the problem to be solved as highlighted in the previous section.

Based on the problems defined, the research questions are identified as follows:

a) *What are the characteristics of current mobile services?*
b) *What are the limitations of current mobile services with regards to open mobile environments?*
c) *Which mechanisms and functionalities are required in introducing a service in order to support unpredictable services with arbitrary names introduced and provided by different service providers?*
d) *What is the current information available to the user during the handover which relates to service information?*
e) *What are the limitations of the current service information with regards to service continuity and service availability?*
f) *How can the service information be improved by taking d) and e) into account?*
g) *Is the existing practical implementation possible today, or if not, what types of technologies are currently missing?*

Research questions *a), b)* and *c)* are based on the first problem statement which focuses on the need for an architecture and framework for unifying mobile services. The next research questions *d), e)* and *f)* are related to the next three problem statements which highlighted on the lack of service information available during the handover (service transfer) support for seamless service continuity. The final question is based on the last problem statement which is on the need of service discovery that can handle new and unpredictable services with arbitrary names introduced and provided by different service providers.

The design-science research guidelines as illustrated in Table 1 are applied for each of the research question.

Table 1 Design-science research guidelines

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Source: Alan R. Hevner, Salvatore T. March, Jinsoo Park, Sudha Ram."Design Science in Information Systems Research". *MIS Quartely* [Research Essay]. (28) 1: 75-105 (2004).

Research question *a)* is applying design-science research guideline no. 5 which is on the research rigor. Rigor is derived from the effective use of the knowledge base which are the theoretical foundations and research methodologies. In this research the theoretical foundations of the current mobile service are studied as a basis for the construction of the design artifact.

For research questions *b)* and *e)* research guideline no. 2 which is problem relevance is applied as the answers from both questions indicate the relevant problem addressed by this research.

Research question *c)* is applying research guideline no. 4 which is on the research contributions as the answer from the question will contribute to the areas of the design artifact – which in this research refers to the new solution for seamless service continuity for the user in an open mobile and a multiple network system environment.

Guideline no. 6 which is design as a search process is applied in research question *d)* where the answer is used to design a mechanism that satisfies laws in the problem environment which in this research refers to the service availability and service continuity.

Next, research question *f)* is applying research guideline no. 1 which is design as an artifact whereby the answer from the question will produce a viable artifact - which in this research refers to the new mobile service architecture and framework, extended functionalities of IEEE 802.21 MIH Services and a new service discovery system which supports the semantics of data.

Finally, research question *g)* is applying research guideline no. 3 which is design evaluation where the utility, quality and efficacy of the design artifact is rigorously demonstrated via well-executed evaluation methods – which in this research is conducted via laboratory experiments.

A summary of the application of design-science research guidelines in each research question is illustrated in Table 2. The research guideline no. 7 which is on the communication of research is not applied in any of the research questions as it is used in the construction methodology as discussed in the next section (1.3.2).

Table 2 The application of design-science research guidelines in each research question

Research Question	Guideline
a) What are the characteristics of current mobile services?	Research Rigor
b) What are the limitations of current mobile services with regards to open mobile environments?	Problem Relevance
c) Which mechanisms and functionalities are required in introducing a service in order to support unpredictable services with arbitrary name introduced and provided by different service providers?	Research Contributions
d) What is the current information available to the user during the handover which relates to service information?	Design as a Search Process
e) What are the limitations of the current service information with regards to service continuity and service availability?	Problem Relevance
f) How can the service information be improved by taking d) and e) into account?	Design as an Artifact
g) Is the existing practical implementation possible today, or if not, what types of technologies are currently missing?	Design Evaluation

Based on the research guidelines, three main activities in the research process are deduced which are:

- *Problem identification* which is deduced from Guideline no. 2 - Problem relevance
- *Solution design* which is by the combination of Guidelines no. 1, 5 and 6 – Design as an artifact, Research rigor and Design as a search process

- *Evaluation* which is from the Guidelines no. 3, 4 and 7 – Design evaluation, Research contributions and Communication of research

The details of activities are as described below:

i) *Problem identification*

   In problem identification activity, the need for service transfer and continuity between disparate but equivalent services is identified. All the handover and service continuity solutions are analysed thoroughly. The limitations of the existing IEEE 802.21 MIH Services are then identified. To provide service continuity, a service discovery is required. The existing service naming and service discovery schemes are also studied and evaluated. Based on the limitations identified, some research questions are derived and set as the foundation for the further research process.

ii) *Solution design*

   In the solution design, the requirements of future mobile services are identified concerning the dynamic changes of components realising a service. Based on the results of the study in the problem identification activity, the requirements of future service naming and service discovery scheme are proposed and described thoroughly. The design of the future service discovery scheme is also conducted and developed. Extensions to the IEEE 802.21 MIH Services are proposed and described in detail. The outputs of this activity then are used to design the mobile service architecture for future mobile services.

iii) *Evaluation*

   In this last activity, the future service discovery system is implemented based on the conceptual model of the design artifact from the solution design activity. Some case studies are presented in use scenarios to show the applicability in practice and laboratory experiments are also conducted on the service discovery system for performance and evaluation testing.

1.3.2. Construction methodology

Since this thesis will involve the implementation of future service discovery schemes and the realisation of mobile service architecture, the Unified Modeling Language (UML) [17] is used to visualize the conceptual models, requirements, designs and deployment architecture.

The diagrams used in each process are:

1. UML use case diagrams – for high-level requirements specifications
2. UML collaboration diagrams – for high-level system interaction specifications
3. UML sequence diagrams – for high-level system process specification

There is no formal testing methodology conducted as the developed system artefacts are only prototypes. However, the system has been sufficiently evaluated with manual testing and the results retrieved are satisfactory and acceptable.

1.4 Contribution of the thesis

The main contributions of this thesis are:

- The specification of the architecture and framework enabling the unification of mobile services.
- The proposed extension of the current IEEE 802.21 MIH Services which enables the support of service transfer and continuity.
- The specification of an open and sound service discovery system, which supports the introduction of any services with arbitrary names by anybody without the need of standardisation.

The thesis resulted in 9 published papers at international conferences with referee and 1 accepted journal paper focusing on the three topics as follows:

- Mobile service architecture and framework
- Service continuity
- Service discovery

Figure 3 illustrates the relationships between the papers and their corresponding area of contributions.

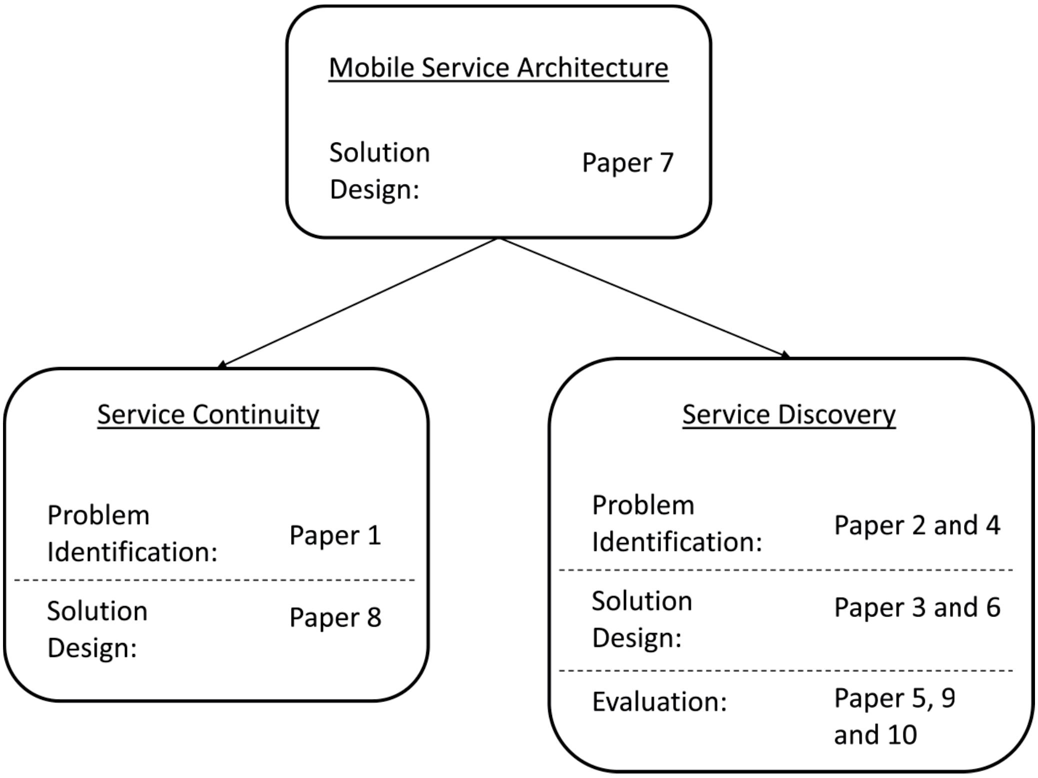

Figure 3 The relationship between the papers and their corresponding area of contribution

The construction methodology is applied in each focus topic and resulted in publications in all the three activities: Problem Identification, Solution Design and Evaluation as follows:

- Mobile service architecture:

  - Paper 7: “A Mobile Service Architecture for improving Availability and Continuity”
- Service continuity:
  - Paper 1: “Enabling Service Continuity on Future Mobile Services”
  - Paper 8: “Service Continuity across heterogeneous networks ”
- Service discovery:
  - Paper 2: “Service Naming in Future Mobile Environments”
  - Paper 3: “Service Discovery in Future Open Mobile Environments”
  - Paper 4: “An Open and Extensible Service Discovery for Ubiquitous Communication Systems”
  - Paper 5: “Service Discovery for Mobile Multi-domain Multi-language Environments”
  - Paper 6: “Service Discovery in Ubiquitous Mobile Computing Environment”
  - Paper 9: “Novel Service Discovery Techniques for Open Mobile Environments and their Business Applications”
  - Paper 10: “Quality-driven Service Discovery Techniques for Open Mobile Environments and their Business Applications”

The limitations found from the problem identification activity in both service continuity and service discovery areas contribute to the design of mobile service architecture.

The author of this thesis is the first author and main contributor to all the 10 papers. However, she received advice, support and assistance from the other co-authors who are her supervisors, her students and her colleagues from the Mobicome project.

1.5 Additional work

1.5.1. Student supervision

Two Master students were supervised by the PhD-candidate during the work of this thesis. The summary of the work done by the students is as follows:

MASTER THESIS: "Service Discovery for Future Mobile Services"

A web-based Future Service Discovery System (FSDS) was successfully developed and implemented by the students using the ASP.NET Framework. The system allows flexibility in introducing services and enables anybody even the end-user to be a service provider. It supports semantics in the service descriptions hence results to a more efficient service matching during the service discovery. Using the Web Ontology Language (OWL) Lite the equivalent or same service can have different names and in different languages and different services may also have similar names. In addition, the service discovery also supports partially matching services, i.e. services that are not exact match of the requested one but have more additional functions. A paper has been published based on this work [18] which also contributes to one of the main topics in this thesis.

1.6 Organization of thesis

The organisation of this thesis is as illustrated in Figure 4. Chapter 1 is the Introduction of the thesis starting with the motivation of this thesis work in Section 1.1. Section 1.2 presents the problem statements and goals of the work. The methodologies explaining how the thesis work is conducted are elaborated in Section 1.3 followed by the contribution of the thesis work in Section 1.4. Section 1.5 describes the additional work carried out during the study and Section 1.6 detailed out the organisation of the thesis. Chapter 2 is the discussion part on the related works and state-of-the-art technology related to this work. A summary of the work is given in Section 3 and is presented by displaying the interrelationships between the papers published. Chapter 4 provides conclusion of the work together with the explanation on further work to be done. Part II contains all the thesis publications and papers submitted for publication.

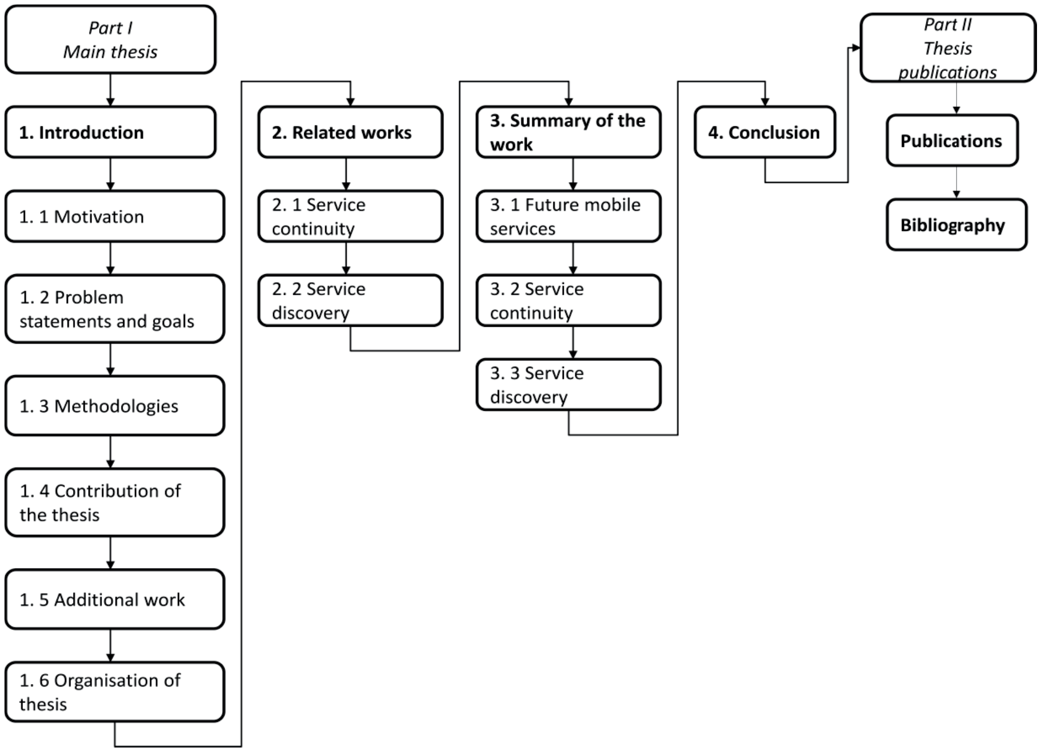

Figure 4 The organisation of this thesis

References are included at the Bibliography the end of the thesis which includes both the references used in Part I as well as the ones in the thesis publications.

2. Related works

Related works are studied thoroughly to see what has been done so far and what are the unresolved issues that require improvements.

2.1 Service continuity

2.1.1. IEEE 802.21 Media Independent Handover (MIH) Services

Current mobile device technology has seen a significant change of the mobile terminal ability, which is not only able to connect to a dedicated wireless network but is equipped with interfaces that allows users to access several wireless access technologies such as Global System for Mobile Communications (GSM), 3rd Generation Wireless Communication (3G), Wireless Local Area Network (WLAN), Bluetooth, etc. In order to ensure the seamless connectivity and facilitate the handover between the heterogeneous networks, the IEEE Working Group has proposed the 802.21 Media Independent Handover (MIH) Services that allows the mobile terminal to detect the communication channels and services that are available in the vicinity.

The IEEE 802.21 has a logical entity called Media Independent Handover Function (MIHF) which is a shim layer in the mobility-management protocol stack of both the mobile node and the network elements. It facilitates the handover decision making based on the upper layer decisions and link selection based on inputs and context from MIH.

Figure 5 illustrates the handover process between different types of networks in the current mobile service architecture supported by the IEEE 802.21 MIH Service.

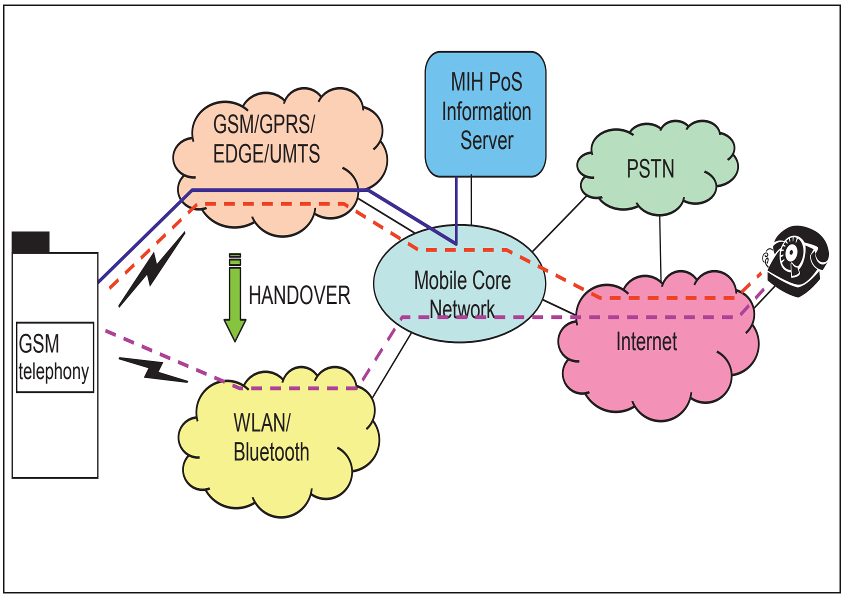

Figure 5 Handover process with IEEE 802.21 MIH Services

Currently without the MIH function, there is no way that the mobile device can detect WLAN coverage because the WLAN interface is usually disabled due to battery drain. To use WLAN, the user has to switch the interface on. Of course it is possible to equip the mobile device with a function that turns on the WLAN interface when the GSM signal quality has deteriorated below a certain level. However, with the today ubiquitous good coverage of GSM the signal quality does seldom fall to the level that can trigger the WLAN switch-on. It is worth noting that the user might want to change to WLAN because of price, performance, security reasons and not because of signal quality. One may conclude that there is no connectivity continuity across heterogeneous access technologies.

With the IEEE 802.21 the mobile device will have continuous access to the MIH function via a SAP on the GSM network. When entering a WLAN area the MIIS will detect it. The MIES will be used to send an event MIH_Link_Detected to the upper layers on the mobile terminal e.g. Mobility manager, which can turn on the WLAN interface.

The mobility manager can also issue the command MIH_MN_HO_Candidate_Query to obtain handover related information about possible candidate networks. It can then make the MIH_MN_HO_Commit to notify the network that a candidate is available. Thereafter, it can initiate the handover by performing the necessary signalling. When handover is completed, the mobility manager will issue a MIH_MN_HO_Complete to indicate the handover has been completed.

However, in the future mobile service architecture, the mobile device will have access to different types of services from different types of networks and they could be different service implementations with different components. For example as illustrates in Figure 6, the mobile phone is moving from a GSM coverage area into a WLAN area which belongs to an IP network operator and does not have any connection to the 3G network. The real challenge is on how to realize the service continuity between these networks. The GSM telephony service is not available in such area but there is an IP telephony service. Even with the assistance of the MIHF, e.g. detection of the WLAN access network, it is not possible to carry out handover because there is no way the mobile phone can perform the necessary signalling to establish a new link to transport the GSM voice traffic. Service continuity cannot be ensured in this way.

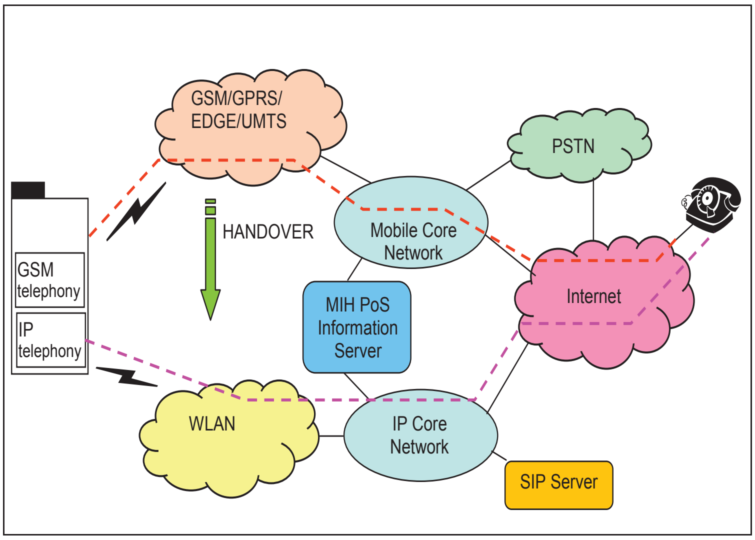

Figure 6 Handover between GSM telephony and IP telephony

The implementation of service continuity for the telephony service in this case requires not only a link change but also the change of the telephony service from GSM telephony service to IP telephony service. To enable a fast "service handover" the following tasks must be accomplished:

- Service discovery
- Service matching
- Initiation of the service client of the mobile terminal
- Registration of the service client to the service server
- Signalling and establishment of new links
- Service switching

The service discovery can, of course be carried out by the mobile terminal but will be more efficient with the assistance of the 802.21 MIHF; something which is not done today.

To elucidate the limitation of the 802.21 MIHF let us consider again the situation of the mobile phone moving from the mobile access network to a WLAN area belonging to an IP network operator. When the mobile phone moves in the WLAN coverage area, the MIIS will detect it. The Media Independent Event Services (MIES) will be used to send an event MIH_Link_Detected to the upper layers on the mobile terminal. The service continuity server can ask for a list of services available in the area which includes the IP telephony service. The service continuity server can perform a service matching and find that the IP telephony service is of type telephony and is equivalent to the GSM telephony service. It is hence possible to do a service handover.

The service continuity server can then initialize the IP telephony client on the mobile terminal. The IP telephony will then proceed with the registration to the SIP server and carry out the necessary signalling to establish a new voice link. To complete the service handover the GSM voice link will be terminated.

Unfortunately, not much is done regarding the architecture of the mobile services that are delivered to the mobile terminal. Hence it is important to understand on how the architecture works and from there it is proposed to have the extension on the current MIH in order to ensure that is supports not only handover between the networks but also service handover (for seamless service continuity).

2.1.2. Future mobile service architectures – Ivar Jørstad (2006)

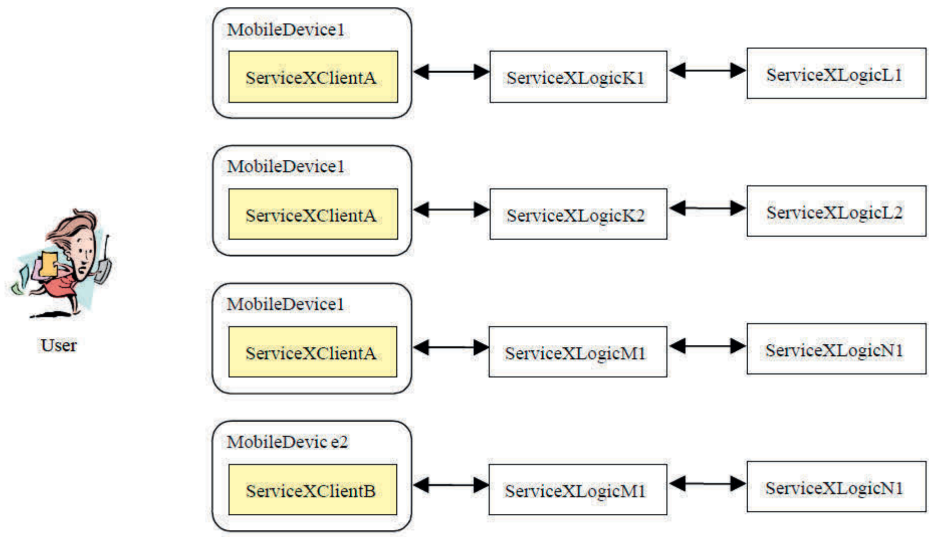

Figure 7 Architecture of a next generation mobile service

I. Jørstad [19] in his PhD Thesis at the Department of Telematics, NTNU has proposed an architecture as shown in Figure 7 which allows dynamic service composition of both pre-defined and ad-hoc service logic components. A service can be realised by a multitude of assemblage variants.

In the first alternative, when the user is using a *MobileDevice1*, a *ServiceX* is realised by *ServiceXClientA*, *ServiceXLogicK1* and *ServiceXLogicL1*.

When the user is moving but using the same mobile device, the *ServiceX* is realised by the same but the logic components *ServiceXLogicK1* and *ServiceXLogicL1* are replaced respectively by new instances *ServiceXLogicK2* and *ServiceXLogicL2* as shown in alternative 2.

In alternative 3, the logic components *ServiceXLogicK1* and *ServiceXLogicL1* are replaced respectively by equivalent implementations *ServiceXLogicM1* and *ServiceXLogicN2*.
When the user changes to another *MobileDevice2*, the *ServiceX* is realised by a new *ServiceXClientB* and the same *ServiceXLogicK1* and *ServiceXLogicL1*.

Jørstad emphasized the importance of being able to realise a service by dynamically discovering, selecting, assembling and executing the service logic components. It is also crucial, with the user's move and location change, to be able to dynamically detect and use new equivalent service logic components instead of the former ones. It is also necessary that the same service logic components can move and change locations.

Furthermore, Jørstad introduces a service continuity layer for the support of service continuity, which consists of the following elements:

1. Monitor: A continuously updated "map" of surrounding networks, domains and hosts
2. Handover Manager: High-level service handover
3. Interoperability Evaluator: Compatibility Matching
4. Service Composition Module
5. Input/Output Redirector

Jørstad's work has brought a new dimension on mobile service architecture by stating that a mobile service can be realised by a multitude of assemblage variants based on both pre-defined and ad-hoc service logic components. Jørstad assumed that there is a consensus on the service definition and functionality but there are multiple different implementations and components.

In this thesis the researcher adopted the idea that a mobile service can be realised dynamically using different components but the assumption of a consensus on service definition is removed. Instead, the researcher introduced a notion of service equivalence, which is left to the user to decide. Consequently, the researcher proposed to realise a mobile service by selecting and using disparate equivalent services. Moreover, instead of introducing a network monitor as proposed by Jørstad the researcher proposed extensions to the IEEE 802.21 MIH.

2.1.3. Mobility management and handover

In the near future there will be the trends of the user being connected almost everywhere and can enjoy whatever services that are available in the network hence making the topic of mobility management and handover as the important issues to be handle.

Currently some effort has been done by few organisations in order to ensure the mobility and handover between the networks. For example, the 3rd Generation Partnership Project (3GPP) has introduced the WLAN interworking [10] which links two or more computers or devices using spread-spectrum or Orthogonal Frequency-Division Multiplexing (OFDM) modulation technology and enables communication between devices in a limited area. This gives users the possibility to move around within a broad coverage area and still be connected to the network. Session Initiation Protocol (SIP) [20] which is specified by the Internet Engineering Task Force (IETF) in RFC 3261 is a signalling protocol which is widely used for controlling multimedia communication sessions such as voice and video calls over IP. It is used to establish sessions in an IP network. It has the key signalling elements which can turn a voice over IP (VoIP) network into a true IP communications network hence contributing to converged services.

IP Multimedia Subsystem (IMS) [21] is an architectural framework introduced by the 3GPP that supports the delivering of multimedia services in the IP network. It uses the SIP to enable easy integration with the Internet. This will help creating a fixed mobile convergence (FMC) by having a horizontal control layer that isolates the access network from the service layer. This will allow the access of multimedia and voice applications from wireless and wired terminals.

All of these technologies support the mobility management and handover between the same type of networks and their concepts can be studied to be used in the thesis, for example methods on discovering new access networks and handover procedures. However the focus of mobility management and handover is only on continuous connectivity between networks while this thesis is on continuity of services.

2.1.4. 3GPP Voice Call Continuity (VCC) and GSMA VoLTE

The 3GPP has conducted studies described in TR 23.806 [22] and has elaborated in TS 23.206 [23] the specifications of the Voice Call Continuity (VCC) which ensure the persistence of a voice call when the mobile phone moves between circuit switched and packet switched radio domains. Voice Call Continuity is ensured but only in two well-known domains where the telephony service is transferred between two known telephony applications, namely GSM voice and IMS.

With the emergence of Long Term Evolution (LTE) paving the way for VoIP over all-IP mobile networks, it is necessary to ensure the continuity of voice calls when a user moves from an LTE coverage area to another without LTE. The GSM Association VoLTE (Voice over LTE) initiative endorses the IMS solution specified by 3GPP. VoLGA (over LTE Via Generic Access) is ensuring continuity by tunnelling GSM voice over IP.

The focus of all these activities is the seamless switching between circuit switched voice and IMS voice, which are two well-defined services provided by mobile operators. This thesis extends service continuity over non-standardised services.

2.1.5. Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN)

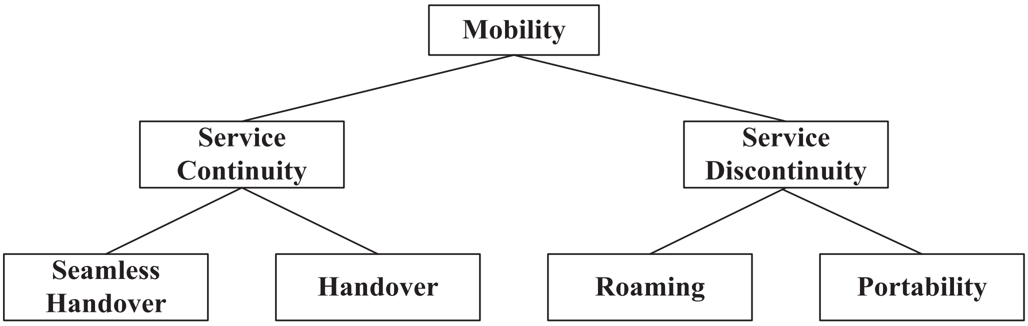

Figure 8 TISPAN's mobility components

In their works of defining the European view of the Next Generation Networking (NGN), TISPAN has studied all the potential service capabilities and requirements [24] and defined the concept of service continuity as a component of mobility [25] as shown in Figure 8.

More specifically, service continuity is defined as "the ability for a user to maintain an ongoing service during mobility".

Although the definition is quite correct TISPAN does associate service continuity to handover as shown in the figure above. This is not always correct because handover is a necessary but not sufficient condition for service continuity.

2.1.6. Mobile services

The W3C Working Group has actively participated in discussing, initiating and producing specifications mobile services. The mobile services can either be:

- Web content that can be accessed via mobile
- Or other services that can be accessed while the user is mobile with the help from the mobile agent

For the first type of mobile services (which is also known as mobile web) the W3C Working Group has introduced XML [26], Resource Description Framework (RDF) [27] and OWL [28] as the guidelines for the Service Providers to offer services via mobile web. XML which is a mark up language as HTML but it is designed to describe data instead of just display them. RDF and RDF Schema describe more meaningful data by having a simple semantic and for generalization-hierarchies of properties and classes. OWL provides even more meaningful data with additional vocabularies for describing properties and classes (e.g. cardinality, equality and symmetry). By having these guidelines it will allow the services to be semantically identified through the descriptive data stored in Web-accessible database or as mark-up within documents and hence let the customer access directly the desired page.

Mobile agent on the other hand making use of the Web technologies to locate data (services) and as well as to perform meaningful tasks on retrieved data.

The described technologies will be used as fundamentals in this thesis to realize service continuity across heterogeneous networks and domains.

2.2 Service discovery

2.2.1. Current Service discovery systems

As mentioned earlier the future mobile environment will consist of multiple heterogeneous networks where a service can be anything with any name and offered by anyone including the end-users. The ultimate requirement posed on the service discovery systems will be as follows:

> *The discovery system must be capable of finding the services requested by the user no matter what the service name or type is.*

By breaking down this requirement into more detailed levels the following requirements are deduced:

- The service discovery must have an open service name and type format meaning that there must not be any restrictions regarding the length of name or the character set.
- The service discovery must have an open service name and type convention meaning that there must not be any rules regarding the selection of the name or type.
- The service discovery must have an open administration meaning that there must not be a central body that assigns and standardizes service names and service types and the service providers are free to choose whatever they like.
- The service discovery must be efficient and sound meaning that it must be capable of finding the requested services in a deterministic amount of time.

Consequently, any service discovery for future mobile environments must fulfill the posted requirements.

Let us now consider the current well-known service discovery systems such as Jini [29], Universal Plug and Play (UPnP) [30], Salutation [31] and Web Services [32]. All of them have one common characteristic which is to have three main components: Service Client, Service Provider and Service Directory or Registry. The Service Client asks the Service Discovery for information about services that are provided by Service Providers.

Each of these service discovery schemes has their own way of offering the service to the user for e.g. Jini uses a proxy called Service Object to implement the service interface and takes care of communication with the remote service. UPnP, on the other hand, will automatically announce their network address and supported device and services types once they are connected to a network. This will allow the user to recognize those types and immediately begin using the device. Salutation uses the Functional Unit (FU) to register when a service is available to the Salutation Manager (SLM). When a user asks the SLM to provide a particular service it will reply to the request by giving the information on the services that the FU offers. Web Service (WS) has a more structured way of introducing/announcing the service via the Universal Description Discovery and Integration (UDDI) acting as a Service Broker. The WS Publisher acting as Service Provider can register its services at the UDDI which can also actively request for a specific service from the UDDI. If the requested service matches with a registered service, the UDDI will send the URL to the WS Subscriber so that it can invoke the service.

All these service discovery systems use a rigid convention and format for service names and types and rely on exact string matching to discover services. Furthermore, these services name conventions and formats are also completely different from each other. Consequently a service name or type from one service discovery system cannot be used in another one and interoperability is completely impossible.

It is hence possible that the current state-of-the art service discovery systems are not satisfying the requirements that the researcher has deduced for the future mobile environments and more innovative ones are needed.

2.2.2. Integrated Semantic Service Discovery – Shanshan Jiang (2006)

Shanshan Jiang [33] from the Department of Telematics, NTNU has proposed an Integrated Semantic Service Discovery which enhances the discovery ability by making use of service ontology. The well-known state-of-the art service discovery systems are using only type, keyword and attribute matching and limit their discovery only to functional properties. Non functional properties are left out.

To remedy the situation Jiang proposed an Integrated Semantic Service Discovery which takes into account both the functional and non-functional properties to achieve accurate and satisfactory discovery results. Ontologies are defined in the Web Ontology Language, OWL [28]. Behavioral semantics are added to WSDL file by associating service functionality related elements with links to OWL-based service ontology. Non-functional properties are specified as QoS parameters and rule-based policies comprising business policies, QoS policies and context policies.

Jiang's achievements have brought useful inputs to this work since the use of semantic has been proposed to improve the Service Discovery. The researcher adopts the same approach in her service discovery in order to enable an open service naming. The Web Ontology Language (OWL) has also been used in the specification of non-functional properties. On the researcher side she uses OWL to describe equivalent services. However, Jiang focused on one limitation of current service discovery, namely the neglect of non-functional service properties while the researcher goal is to develop a service discovery which is capable of finding services having arbitrary names in any format given by any service provider.

2.2.3. Other semantic service discovery systems

Many research efforts have been done on semantic service discovery and lots of research initiatives proposed especially on enhancing discovery mechanisms and solve the problem of finding the right service to the user in an acceptable time.

Efficient Routing Grounded on Taxonomy (ERGOT) [34] for example, is a semantic-based system for service discovery in distributed infrastructures. It is a system that combines Distributed Hash Tables (DHTs) [35] and Semantic Overlay Networks (SONs) [36] to enable semantic-based service discovery in distributed infrastructures such as Grids and Clouds. ERGOT takes advantage of semantic annotations that enrich service specifications in two ways whereby the services are advertised in the DHT on the basis of their annotations, thus allowing establishing a SON among service providers and annotations enable semantic-based service matchmaking, using a novel similarity measure between service requests and descriptions. It is built upon three main elements which are a DHT protocol - used to advertise semantic service descriptions annotated using concepts from an ontology, a SON - which enables the clustering of peers that have semantically similar service descriptions and a measure of semantic similarity among services descriptions - which overcomes the exact-term search limitations of DHTs. This measure relies on semantic annotations of service descriptions expressed in Semantic Annotated WSDL (SAWSDL) and the ranking mechanism is based on semantic similarity.

A novel OWL-S (formerly DAML-S) based semantic service discovery system is proposed by Le Duy Ngan et. al [37] for dynamically discovering complex constraint-based services. It is based on representing complex service constraints as Semantic Web Rule Language (SWRL) rules and using a rule engine for matchmaking and handling dynamism via a real-time ontology population and reasoning infrastructure. It supports descriptions with complex constraints and matching of service providers and requesters dynamically.

UbiSearch is a semantic service discovery network for large-scale ubiquitous computing environments [38]. A semantic service discovery network in the semantic vector space is proposed where services that are semantically close to each other are mapped to nearby positions so that the similar services are registered in a cluster of resolvers. Using this mapping technique, the search space for a query is efficiently confined within a minimized cluster region while maintaining high accuracy in comparison to the centralized scheme. It supports scalable semantic queries with low communication overhead, balanced load distribution among resolvers for service registration and query processing and personalised semantic matching.

An Ontology-enhanced cloud service discovery system (CSDS) [39] aims to support the Cloud users in finding a Cloud service over the Internet. The CSDS interacts with Cloud ontology to determine the similarities between and among services. The Cloud Service Reasoning Agent (CSRA) determines the relations of Cloud services using three service reasoning methods which are similarity reasoning, equivalent reasoning and numerical reasoning.

Ying Zhang, et al. proposed a hierarchical and chord-based semantic service discovery system for the universal network (UniNServ) [40]. It uses OWL-S (Web Ontology Language for Services) to describe services and adopts Chord [41] as a distributed lookup protocol. Besides, UniNServ uses three types of ontologies to perform automatic semantic service discovery with QoS through exploiting the logical relationships within the services. The QoS measurements is appended to OWL-S and called as OWL-QoS.

MEMORY [42] is a matrix-based efficient semantic web service (SWS) discovery system which does ontological pre-reasoning and holds the reasoning results in matrix forms in service publishing phase, so that it can transfer the load of semantic reasoning from service query to service publication and perform fast matching during service discovery. SWS discovery exploits service semantic metadata to reason on the compatibility and functionality information of services. The techniques of SWS discovery can be classified into logic based approaches, non-logic based approaches, and hybrid approaches. Logic-based approach describes services with service description languages such as OWL-S or WSMO and does service matching based on ontology reasoning. Non-logic based approach uses method without ontology reasoning such as clustering to exploit semantic information of service parameters. And hybrid approach combines means of both. It adopts logic based approach and accepts services described with OWL-S or SAWSDL. The service matchmaking is using a three-phase SWS matching strategy, which includes parameters-based service filtering, matrix-based functional matching and QoSbased non-functional matching. In order to conduct fast matching during service discovery, MEMORY prereasons on the semantic information during service publishing and keeps the reasoning results in matrix forms.

A layered architecture of semantic service discovery system is proposed by V. Kaewmarin, et al. which incorporates a search crawler as a core component for discovering services residing on the provider websites [43]. The search crawler can operate in multi-threaded environments to enhance the capability of discovering a number of distributed Web services simultaneously, as well as in various UDDI registries. In order to enrich the service discovery in a semantic manner, the Web services descriptions returned from the search crawlers are thus modelled into a machine-processable representation language such as OWL-S. The system conformed to the proposed architecture will provide the flexibility and extensibility to accomplish complex Web service requests that meet user-specified functional requirements.

A two stages method of semantic service discovery is proposed in [44] whereby the service is divided into two classes - core service description registered at core service library (CSL) and assistant service which is registered at assistant service library (ASL). The service requirement is send to two stages service discovery module by the service requester and the module discover target service through two stages: 1) in core service discovery stage - find target core service according to similarity of service description between service requirement and core service; 2) in assistant service composition stage - form target composite service through composing target core service with related assistant services.

Most of the above mentioned semantic service discovery systems main objective is to deliver and find the right service for the user in an acceptable time. Thus it has become an important factor for an efficient service discovery system but the researcher proposes a more comprehensive and novel service discovery system in open mobile environments which should fulfil the requirements as stated in Section 2.2.1. In short, the semantic service discovery system proposed in this thesis is capable of finding services having arbitrary names in any format given by any service provider.

Related works

3. Description of the work

This chapter describes the work that has been done during the doctoral study. Summarised in the 10 papers published at international conferences the work addresses three main issues as follows:-

- Future mobile services architecture and framework
- Service continuity
- Service discovery

These three issues will now be successively described in the following sections.

3.1 Future mobile services architecture and framework

The work described in this section is covered by paper 7.

Advances in microelectronics and wireless access technologies allow mobile terminals to have multiple simultaneous connections to several networks. Unfortunately, this capability is not yet fully exploited to provide the users with a better and more reliable set of services from multiple networks. Indeed, a mobile terminal, as its name says, is a device terminating a network. It is still a device intended for the delivery of services managed by the network such as telephony, SMS, MMS, etc. The architecture of the mobile services is rather simple. In fact, a mobile service is realised as either a stand-alone application executing the mobile phone or the SIM card, or a thin client-server application. Even the complex GSM voice telephony service can also be modelled as the client-server. This is achieved by omitting the separated switching function in accordance to the Intelligent Networks concept[2] and focusing only on the service function. Next, a call consisting of two parties A and B can be split into two half calls and each half call can finally be modelled as client-server i.e. consisting of a client and a server part. Consequently, the GSM voice telephony

[2] ETSI EN 301 931-4 V1.1.2 (2001-07) Intelligent Network Capability Set 3 (CS3); Intelligent Network Application Protocol (INAP);Protocol specification;Part 4: SDLs for SCF-SSF interface

service can then be modelled as a client part realised by hardware and firmware on the mobile phone and a server part represented by MSC/SCP[3] node.

A special variant of the thin client-server model which is commonly used is the "browser model" used to access services at web sites. To improve the service offering and the user experience the researcher proposes introducing a new service architecture, which allows dynamic composition of a service using components from different networks as the user is moving. The mobile phone from a peripheral device will become the central device that selects not only services but also service components.

3.1.1. Clarification of the service concept

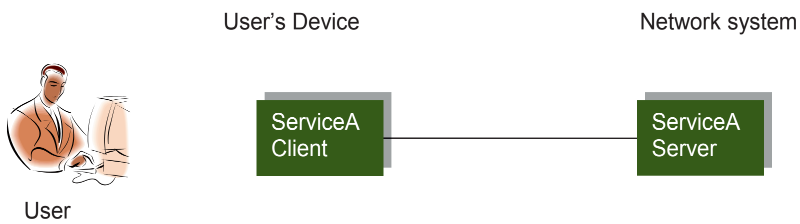

Figure 9 A generic service architecture

To avoid confusion it is necessary to have an unambiguous definition of the service concept used in this thesis.

Service is a concept which is broadly used by numerous communities and hence quite confusing. In this paper, the definition of service is adopted from OASIS Reference Model for Service Oriented Architecture [4] as follows:

"A service is a mechanism enabling the end-user's access to one or more capabilities"

The reason for choosing this definition lies in the fact that it focuses on the user and what capabilities or functions he/she can get from the network while the other definitions

[3] MSC: Mobile Switching Center – SCP: Service Control Point

emphasize quite often on the structure and composition of the service. This definition is fundamental in this thesis since it paves the way for the realisation of generic services and service continuity based on service components developed by different parties but perceived as equivalent by the user. Indeed, it is important for the user to have an uninterrupted conversation on the phone and it is clearly less important that the call is provided by a GSM voice telephony or any arbitrary IP telephony.

In a mobile environment, a service can be conceptually realised by two components.

- A server on the network, which can be both hardware and software and distributed over multiple network elements.
- A client on the user's device which can be:
    - Generic client such as the Web browser that is capable of collaborating with any service server.
    - Specific or dedicated client that could collaborate only with some specific service servers. Typical examples are Web applications or in short, Web apps.

In this thesis, the focus is only on specific clients and how to provide service continuity on these clients.

3.1.2. Requirements on future mobile services

A. Handling of equivalent services

In a multi-network environment a service can be implemented using different architecture and technologies for different network systems resulting in different user interfaces. For example, the telephony or voice service can be implemented as GSM voice or an arbitrary IP telephone call such as chIPhone. Seen from the user these two services are equivalent since they both offer human-to-human voice communication. However, they are currently considered as completely unrelated. When the mobile device is moving connections to the network systems are also changing, a service currently in use e.g. GSM voice may no longer be available and another one e.g. chIPhone may appear. Unfortunately, the mobile device is not aware of the availability of this equivalent service in order to use it. Of course, the detection and swapping to the pop up service is not only motivated by the availability but also other factors like ease of use, performance, security, price, etc. Furthermore, when a service e.g. voice service is ongoing it is desirable that it is not interrupted. Service continuity is another required characteristic. Seen from the user it is valuable to be able to manage this sort of equivalent services in a uniform and simple way to improve service availability and preserve service continuity.

B. Providing location-based services

When travelling, the mobile user will want to have access to the same services such as taxi, hotel, restaurant, flight booking, etc. which enable the necessary booking, modification, cancellation, etc. when needed. At first glance, the mentioned services look the same everywhere but a closer look reveals the opposite. Indeed, when being in Trondheim, the user will need to communicate with the Trondheim taxi service and not the one in London. It is also desirable that the generic taxi service switches to the right service according to the location automatically without the intervention of the user. In fact, it may be sometimes difficult for the user to find the taxi service in a country, e.g. China, Cambodia, etc. where a different language is used. Seen from the user, it is valuable to seamlessly access the right service based on the location.

C. Offering personalised service

As a human being, each user will have some preferences related to the services and it will be very convenient if these preferences are remembered and dynamically re-applied to every equivalent service in use. In addition, the personal data such as personal details, address, phone numbers, contact list, etc. should also be shared between all the services.

Seen from the user, it is valuable to be able to use the service in a personalised way and personal data can be handled at one place.

3.1.3. A mobile service architecture enhancing availability and supporting continuity

Let us start now with the elaboration of a mobile service architecture enhancing availability and supporting continuity. As design tools, the researcher selected simplified versions of three diagrams of UML [45] namely collaboration and sequence diagram to capture the behavioural aspects of the proposed architecture. Although UML has more diagrams we found the three diagrams sufficient for our purpose which is an overall understanding of the architecture and not the generation of programming code. For the implementation, since our system encompasses several heterogeneous environments we have to consider and develop codes for multiple SDKs (Software Development Kit).

When the user is moving the service A as shown in Figure 10 currently in use may cease to be available. Fortunately, a service B which is perceived as equivalent to service A by the user is emerging. The user will probably want to use service B instead of service A and the replacement should preferably be carried out in smooth and seamless way. Until now this is not possible since the two services are considered as completely different and there is no mechanism enabling the service transfer.

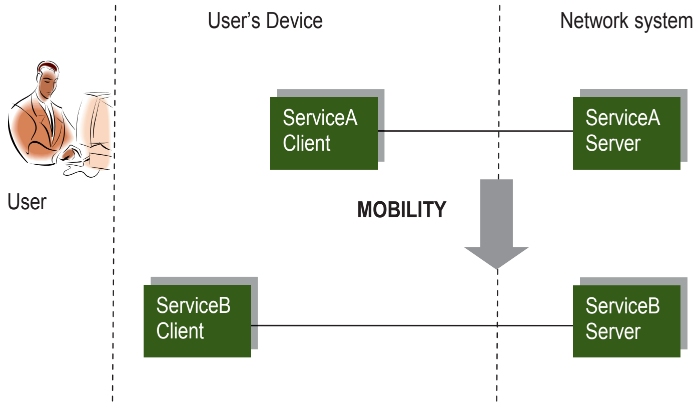

Figure 10 Services in a mobile environments

To enhance the service availability and continuity, the researcher proposed to introduce a new entity called **Generic Service** on the user's device as shown in Figure 11. The Generic Service is equipped with a user interface that enables receiving commands from the user and to generate output. The Generic Service can be implemented as a native application for iOS[4], J2ME, Android, etc. or as a Web application, which can be downloaded and installed by the user. When the user dials and initiates a call on the Generic Service depending on their availability either service A or service B will be invoked. In Paper 8, the proposed solution is to let the Generic Service invoke directly Service A Client or Service B Client. This requires that the Generic Service must have sufficient knowledge to interact with Service A or Service B, which could be a limitation. To remove this limitation an enhanced architecture is proposed in this section. Two objects, namely **Continuity Agent** on the user's device and **Continuity Server** on the network system side are introduced to assist the Generic Service in interacting with Service A and Service B.

In this solution, no direct interaction between the Generic Service and the Service Client is required. The Generic Service is a very light software client with the main role of communicating with the user. No requirement is put on the Service A or Service B.

[4] Apple iOS formally iPhone OS is the operating system of iPhone

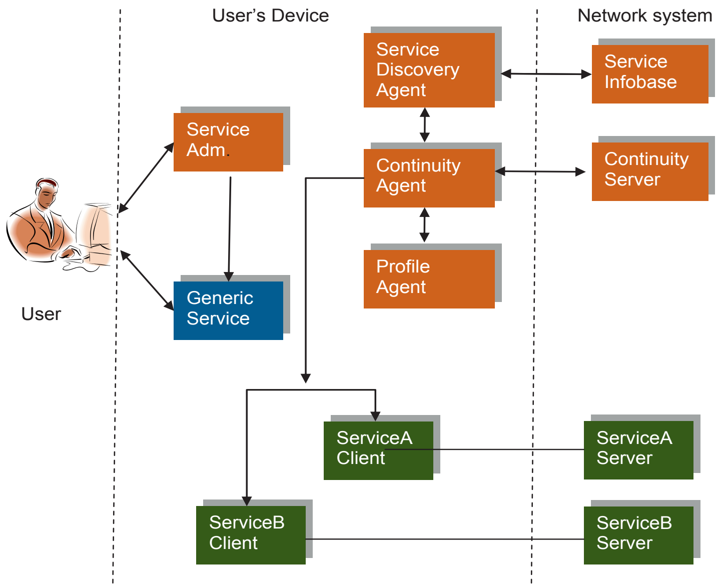

Figure 11 A Service Continuity architecture

Now, the challenge is to know when and how to switch between service A and service B. Again, the **Continuity Agent** is used to assist the Generic Service in the switching process between services. It will perform the necessary initialisation and configuration such that the replacing service can continue from the point the former service stops. In the case where a new service client is required on the mobile phone, the Continuity Agent should also be capable of downloading, installing and initiating this new client. The Continuity Agent gets assistance from the **Continuity Server** on the network system to carry out seamless transfer of synchronous services like telephony. The Continuity Server is a media server realized by the Snowshore server, now called Dialogic IP Media Server [13]. This is used to create and bridge multiple RTP media streams. For operability between the GSM network and the IP network, a Gateway has to be introduced to convert the signalling between SIP and SS/7 and to transcode the media streams traversing the IP network and the IP network.

The Continuity Server needs assistance to find dynamically the available services at every place. An entity called **Service Discovery Agent** is introduced to perform the discovery of available services in collaboration with an entity called **Service Infobase** in the network system. The Service Infobase is a service registry that contains information about all the services available at a location. It has also capabilities to provide a complete list of available services to carry out matchmaking to find the specific services requested by the Service Discovery Agent. The service information stored in the Service Infobase should include details about service equivalence as specified in [46] which allows the discovery of equivalent services. The Service Infobase is also equipped with an open interface enabling all the service providers, including the end-users to register the services they are providing. The Service Infobase provides also an *Event service* that the Service Discovery Agent can subscribe to receive notification when networks and services emerge due to the user's mobility. The Service Infobase can also be realised as an extension of the IEEE 802.21 MIH and more details will be given in later sections.

At service switching, to avoid cumbersome repetition, it is convenient for the user if the preference settings and personal data can be shared among the equivalent services. For that the **Profile Agent** is introduced to store and administer the user profile. It has interface allowing the Continuity Agent to fetch the necessary data to configure the service and to save all the changes done by the user when using the services.

The proposed architecture as shown in Figure 11 is a high level functional architecture which enhances availability and continuity for both asynchronous mobile services like Web browsing, messaging, chat, etc. and synchronous mobile services like telephony. It is worth noting that Figure 11 depicts a high level architecture and all the detailed interactions are omitted for clarity sake. In the next section, the proposed architecture will be elaborated further and more details will be added to show the support for a typical synchronous service, namely telephony.

Service continuity support for telephony

Let us now consider telephony which it is most popular synchronous service. A Generic Telephony service is introduced to offer a unified and continuous voice communication service. In order to identify interfaces and operations for the Generic Telephony service and for other continuity entities like Continuity Agent, Continuity Server, Profile Agent, etc. the sequence diagram for the service transfer is elaborated.

The whole service transfer process for A-party is illustrated in Figure 12.

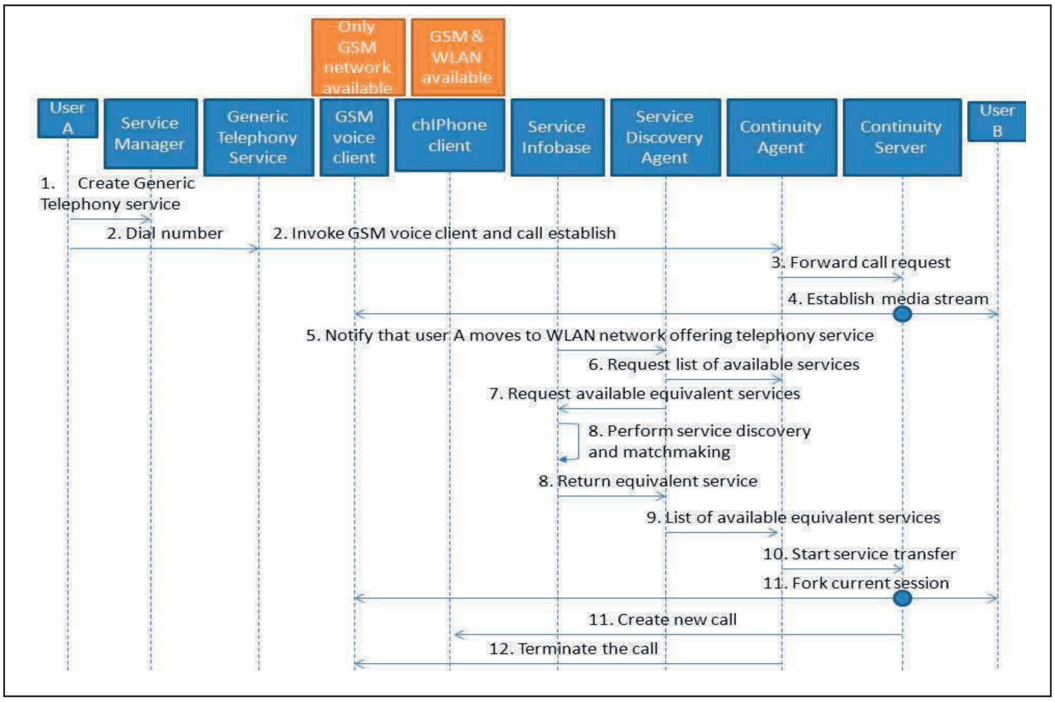

Figure 12 Sequence diagram for the service transfer process – A party

1. The user A communicates with the *Service Manager* to create a *Generic Telephony service*, which incorporates both *GSM voice* service or *chIPhone*, a SIP-based IP telephony service. The GSM voice client is factory installed at while *chIPhone* is downloaded and installed by the user.

2. The user A dials a number on the Generic Telephony service at a place where only GSM voice is available. The *Generic Telephony service* sends a call request to the Continuity Agent.
3. The Continuity Agent forwards the call request to the Continuity Server.
4. The Continuity Server establishes a media stream between the GSM voice client and B-party enabling a conversation between them. This is done by sending INVITE message towards the GSM/PSTN networks via Media Gateway, which converts SIP into SS7. It is worth noting that in this context the Media Gateway to GSM can be considered as the GSM voice server part. For simplicity sake the Media Gateway is not shown on the sequence diagram.
5. The user A is moving to a place where there is WLAN network offering an IP telephony service. The *Service Infobase* sends notification to the *Service Discovery Agent.*
6. The *Service Discovery Agent* requests the list of services demanding availability and continuity from the *Continuity Agent*, which contains *GSM voice* and *chIPhone.*
7. The *Service Discovery Agent* requests the *Service Infobase* to find all the available services that are equivalent to *GSM voice* and *chIPhone.*
8. The *Service Infobase* carries out the service discovery and matchmaking, and returns to the *Service Discovery Agent* a list containing a service *sIPhone*, which is SIP-based telephony and equivalent to *chIPhone*.
9. The *Service Discovery Agent* hands the list of available services over to the *Continuity Agent* for the initiation of the service transfer.
10. Since the chIPhone client is already a SIP client, the *Continuity Agent* does not have to download and install any SIP client but only initialize the *chIPhone client*. The *Continuity Agent* requests the *Continuity Server* to start the service transfer.
11. The *Continuity Server* forks the current session between a B-party and the *GSM voice* client and creates a new call leg towards the *chIPhone client*.

12. When the *chIPhone client* is ready and running the Continuity Agent terminates the call leg with the *GSM voice client*. This is a typical soft handover since there is a short overlapping period where both clients are operating.

For B-party the service transfer is realised by the same continuity entities i.e. Generic Telephony Service, Continuity Agent, Continuity Server, Profile Agent, etc. It is worth noting that A-party and B-party are symmetrical i.e. have the same capabilities because both parties should be able to make and receive calls.

The whole service transfer process for B-party is illustrated in Figure 13.

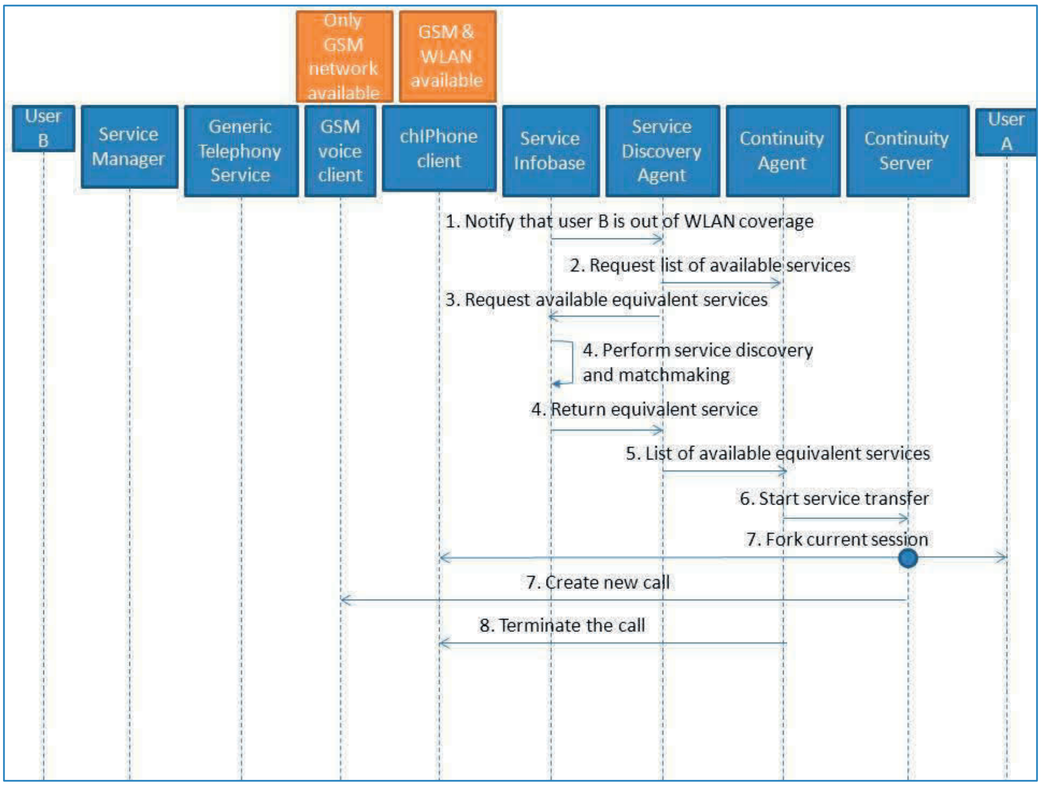

Figure 13 Sequence diagram for the service transfer process – B party

Assume that user B is engaged in a telephone conversation using an IP telephony service called IPphone. It is worth noting that a corresponding process will apply for the case of GSM voice service or other IP telephony services.

1. The user B is moving to a place where there is no WLAN network The Service Infobase sends notification to the Service Discovery Agent.
2. The Service Discovery Agent requests the list of services demanding availability and continuity from the Continuity Agent, which contains IPphone.
3. The Service Discovery Agent requests the Service Infobase to find all the available services that are equivalent to IPphone.
4. The Service Infobase carries out the service discovery and matchmaking, and returns to the Service Discovery Agent a list containing only GSM voice service.
5. The Service Discovery Agent hands the list of available services over to the Continuity Agent for the initiation of the service transfer.
6. Since the GSM voice client is inherently installed in the mobile phone, the Continuity Agent does not have to perform any installation. The Continuity Agent requests the Continuity Server to start the service transfer.
7. The Continuity Server forks the current session between a A-party and the IPphone client and creates a new call leg towards the GSM voice service client.
8. When the GSM voice client is ready and running the Continuity Agent terminates the call leg with the IPphone client.

3.1.4. Implementation and verification of service continuity

A proof-of-concept prototype demonstrating the proposed service continuity architecture is successfully developed and implemented using web-service and mobile phone technology. It will be further discussed in Section 3.3.4 and 3.3.5.

The Continuity Server is realised by the Voice server implemented by the EUREKA Mobicome[5] project [12], which is focusing on providing service continuity across multiple devices. For example, the user is talking to the mobile phone while walking to his office. When arriving at his office, he transfers the call to his multimedia PC and continues the conversation while taking notes since he does not have to hold the phone anymore. The Voice server, later renamed to Media server is realized by the Snowshore server, currently called Dialogic IP Media Server which is capable of creating and bridging multiple RTP media streams and hence enables the establishment of a new calling leg without disrupting the call. The Dialogic IP Media Server is connected to the PSTN and GSM networks via Media Gateway which converts the SIP signaling to SS7 and RTP streams into TDM (Time Division Multiplexing) streams. It is worth noting that transferring a call from GSM voice client on one device to an IP telephony client on another device is conceptually similar to transferring a call from a GSM voice client to an IP telephony client residing on a same device. However, the latter case is not yet implemented due to the unavailability of open source IP telephony client at the time this thesis work was carried out.

A simplified version of the **Service Infobase** supporting the discovery of equivalent services in different languages has been successfully implemented [18]. In addition, the discovery of partially matched services; i.e. services satisfying only a subset of the functionality has been realised.

[5] The EUREKA Mobicome project (2007-2010) is aiming at providing fixed-mobile convergent IMS environment with the participation of Telenor, Telefonica, Ericsson, Huawei, Linus, Ubisafe, HiQ, Oslo University College, Polytechnical University of Madrid and Blekinge Institute of Technology

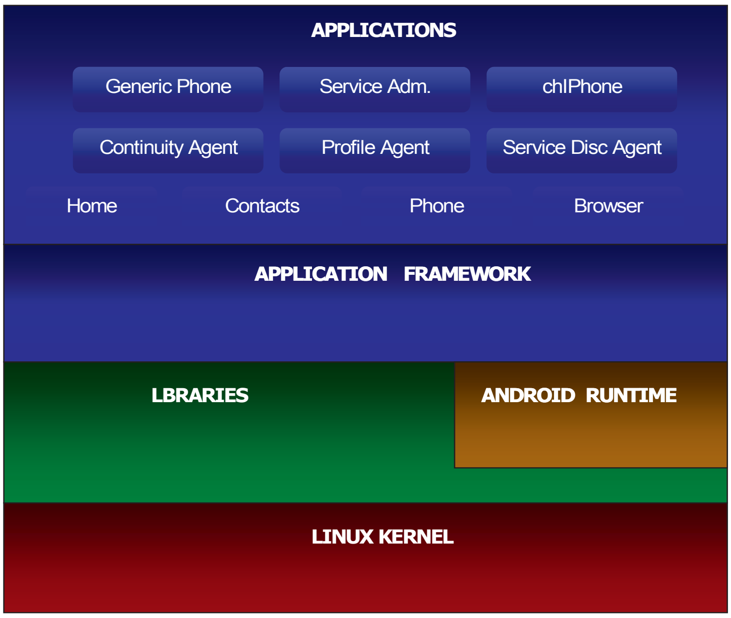

Figure 14 A Service Continuity implementation on Android

The development of entities on the mobile device is carried out on Android [47], which is a software stack for mobile devices that includes an operating system, middleware and key applications. As shown in Figure 14 the service continuity entities are introduced as applications. The *Generic Phone* will switch via the Continuity Agent between the native *Phone* application and *chIPhone*, a SIP client. The *Profile Agent* will communicate with the native *Contacts* application to provide proper configuration and contacts for *Phone* and *chIPhone*.

3.2 Service continuity

The work described in this section is covered by paper 1 and 8.

Service continuity is defined as “the ability for a user to maintain an ongoing service during mobility”. As stated in the previous section service continuity is a central element for future mobile services. This section provides an analysis of service continuity and proposes a realisation of service continuity by extending the IEEE 802.21 Media Independent Handover (MIH) Services.

3.2.1. Analysis of service continuity

A future mobile service will be realised by several equivalent implementation alternatives using different technologies and having different service components. When the mobile device is moving it might be necessary to change the implementation alternative and its components due to different reasons such as air link availability, service implementation availability, quality of service, security, pricing, etc.

An example of service change is the case of telephony service which is realised first by GSM telephony and then IP telephony. To ensure service continuity, it is necessary that a change of service implementations is carried out in addition to a change of connected links.

More generally, the following statements can be formulated:

- A mobile service can be realised by one or more service implementations
- A service implementation consists of one or more service components
- Service continuity is ensured by both a change of connection links and a replacement of one service implementations with another one

From the statements three cases were derived which are:

- Case 1: Change of links only (see Figure 15)
- Case 2: Change of links and service logic components in the network (see Figure 16)
- Case 3: Change of links and service logic components both in the client and network (see Figure 17)

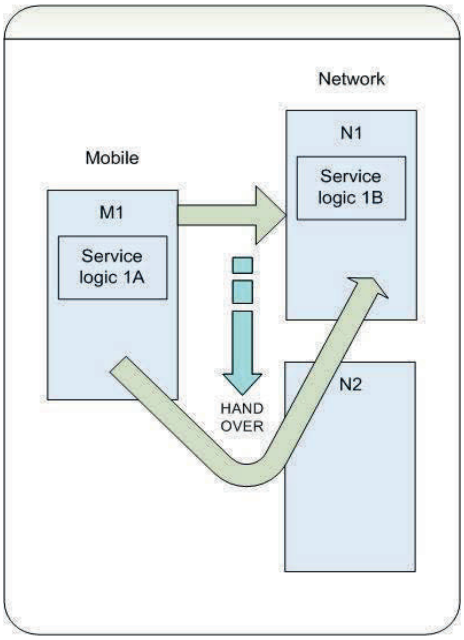

Figure 15 Case 1: Change of links only (User moves to a new network connection without changing the existing mobile service logic components)

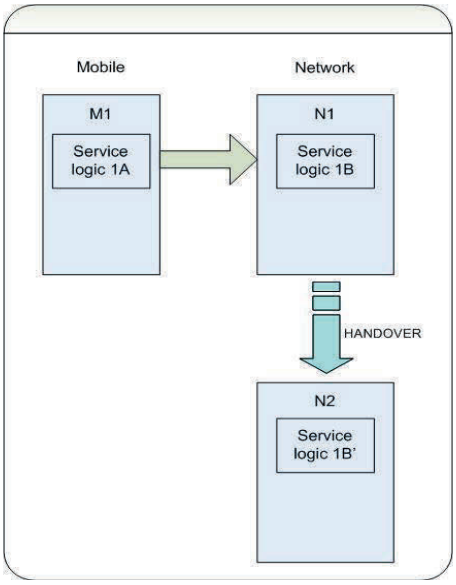

Figure 16 Case 2: Change of links and service logic components in the network (User moves to a new network connection which has different equivalent service logic components)

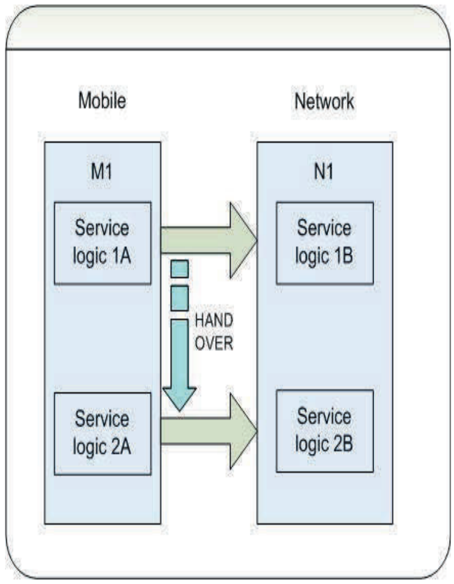

Figure 17 Case 3: Change of links and service logic components both in the client and network (User moves to a new network connection which has different equivalent service logic components and change the mobile service logic components)

As depicted from the figures above, the user can still enjoy the service (in this example, a telephony service) provided that it is realised by the equivalent service logic components in both the mobile and the network side.

In order to support service continuity, the future mobile service environment must support the dynamic service composition, i.e. allowing a service to be composed dynamically by equivalent service logic components. Indeed, there is a need to ensure that the realisation of the service can be done by dynamically discovering, selecting, assembling and executing the service logic components. In order to realise the service, there is a need for information about service components such that identification and matching of service components can be done.

To determine the semantic equivalence of two services and the semantic equivalence of the service components, there must be a precise definition of services, service components and service ontology. As explained in [48], the semantic equivalence of two services S_1 and S_2 is determined by:

1. Semantic equivalence of the service concept
2. Semantic equivalence of the interfaces

And in this example S_1 ***SE*** S_2 which states that S_1 is semantically equivalent to S_2.

Finally to use the service components, the syntactic properties of the services should also be determined.

3.2.2. Realisation of service continuity by extending the IEEE 802.21

A. Short introduction about the IEEE 802.21 Media Independent Handover Function (MIHF)

Seamless handover in a horizontal network (i.e. change from one Base Station to another one Base Station in GSM network telephony) is a well-known technology. Seamless handover in heterogeneous network (i.e. changing from GSM network to WLAN) is a more challenging issue since it requires continuous scanning of a broad spectrum of frequencies using different coding schemes, which is very power consuming.

With the introduction of IEEE 802.21 Media Independent Handover Function (MIHF), the mobile device can perform handover while consuming minimum energy. Its interface, Service Access Points (SAPs) enables the mobile device to have continuous access to the network. Information about neighboring networks is shared through the MIHF, a shim layer in the mobility-management protocol stack of both the mobile node and the network elements. It helps mobile device to discover, characterize and select a network within their current neighborhoods by exchanging information and defining commands and events to assist in the handover decision making process.

Unfortunately, seamless handover is not sufficient to ensure service continuity because in many cases, due to mobility not only the connection link must be changed but also the components realising a service. To give adequate assistance to the dynamic composition of a service there is a need for extensions in the MIHF.

Figure 18 shows the components of MIH and how its function.

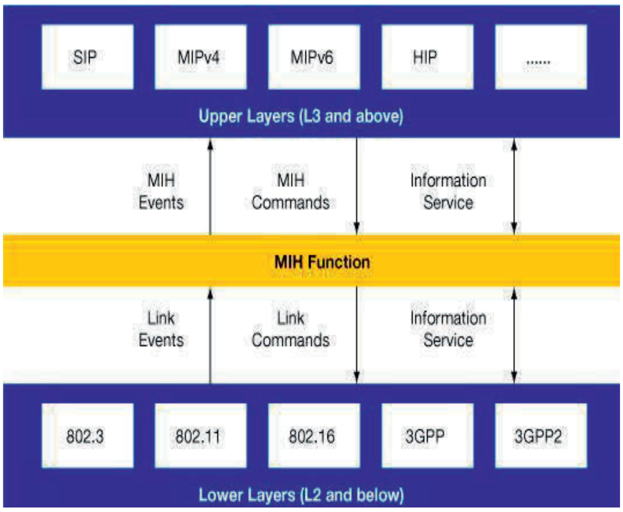

Figure 18 IEEE 802.21 MIH Architecture

The MIHF consists of three major services as follows:

- **Media Independent Event Service (MIES)**: consists of functions to detect events and deliver triggers from local and remote interfaces to the upper layers
- **Media Independent Command Service (MICS)**: consists of functions to provide a set of commands for MIHF users to control handover relevant link states

- **Media Independent Information Service (MIIS)**: consists of functions to provide details on the characteristics and services provided by the securing and surrounding network

To illustrate the role of IEEE 802.21 in handover let us consider the case when the mobile phone is moving from a GSM network to WLAN.

Currently without the MIH function, there is no way that the mobile device can detect WLAN coverage because the WLAN interface is usually disabled due to battery drain. To use WLAN, the user has to switch the interface on. No service continuity is supported.

With the IEEE 802.21 the mobile device will have continuous access to the MIH function via a SAP on the GSM network. When entering a WLAN area the MIIS will detect it. The MIES will be used to send an event MIH_Link_Detected to the upper layers on the mobile terminal e.g. Mobility manager, which can turn on the WLAN interface.

The Mobility manager can also issue the command MIH_MN_HO_Candidate_Query to obtain handover related information about possible candidate networks. It can then make the MIH_MN_HO_Commit to notify the network that a candidate. Thereafter, it can initiate the handover by performing the necessary signaling. When handover is completed, the mobility manager will issue a MIH_MN_HO_Complete to indicate the handover has been completed.

The handover process is as illustrated in Figure 19.

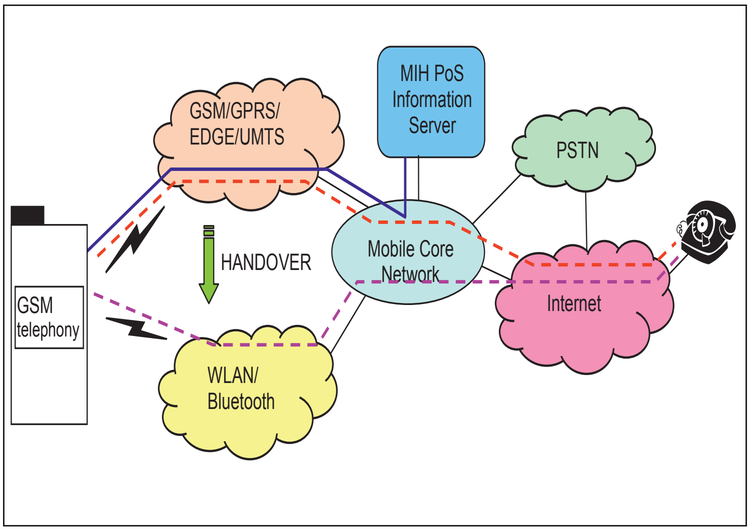

Figure 19 Handover process with IEEE 802.21 Information Service

B. Limitations of the 802.21 MIH

As described in the previous section, the 802.21 Media Independent Handover Function (MIHF) does successfully support handover between heterogeneous access networks that are incorporated in the same core network as specified by 3GPP in [10]. The access networks may belong to the same network operators or different ones. Service continuity is obtained with a link through the new access network.

It is worth noting that the necessary condition for success is that the same service implementation is used across all heterogeneous access networks. In the previous example, GSM telephony service is used both in the mobile access networks and in the WLAN network.

It is more challenging to realise service continuity when the mobile phone is moving from a GSM coverage area into a WLAN area which belongs to an IP network operator and does not have any connection to the 3G network. The GSM telephony service is not available in such area but there is an IP telephony service. Even with the assistance of the MIHF, i.e. detection of the WLAN access network, it is not possible to carry out handover because there is no way the mobile phone can perform the necessary signalling to establish a new link to transport the GSM voice traffic. Service continuity cannot be ensured in this way.

The implementation of service continuity for the telephony service in this case requires not only a link change but also the change of the telephony service from GSM telephony service to IP telephony service. To enable a fast "service handover" the following tasks must be accomplished:

- Service discovery
- Service matching
- Initiation of the service client of the mobile terminal
- Registration of the service client to the service server
- Signalling and establishment of new links
- Service switching

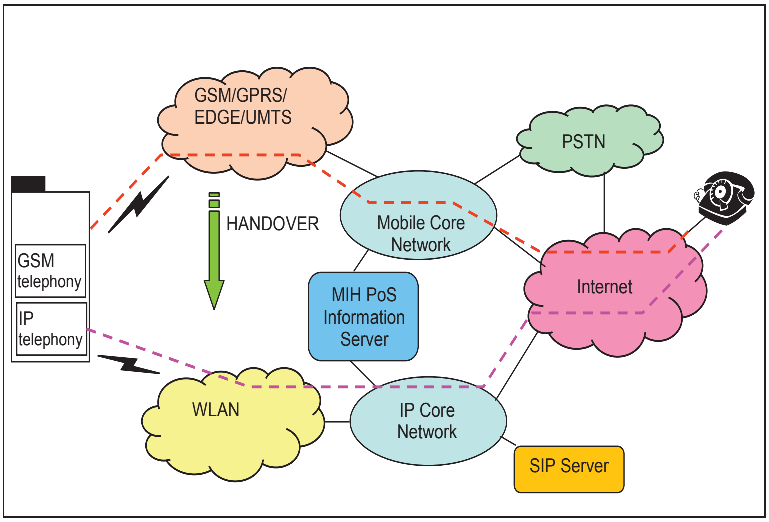

Figure 20 Handover between GSM telephony and IP telephony

The service discovery can, of course be carried out by the mobile terminal but will be more efficient with the assistance of the 802.21 MIHF; something which is not done today. To elucidate the limitation of the 802.21 MIHF let us consider again the situation of the mobile phone moving from the mobile access network to a WLAN area belonging to an IP network operator.

When the mobile phone moves in the WLAN coverage area, the MIIS will detect it. The MIES will be used to send an event MIH_Link_Detected to the upper layers on the mobile terminal. The service continuity server can ask for a list of services available in the area which includes the IP telephony service. The service continuity server can perform a service matching and find that the IP telephony service is of type telephony and is equivalent to the GSM telephony service. It is hence possible to do a service handover.

The service continuity server can then initialize the IP telephony client on the mobile terminal. The IP telephony will then proceed with the registration to the SIP server and carry out the necessary signalling to establish a new voice link. To complete the service handover the GSM voice link will be terminated.

C. Proposed extensions to the MIH handover procedure

In order to implement the extension necessary for the support of service continuity in the case of handover between two different but equivalent service let us consider the Mobile-initiated handover procedure and identify the steps where addition or modifications are to be done.

The Mobile-initiated Handover Procedure operates as follows:

- **Step 1:** Mobile Node is connected to the serving network via Current Point of Service (PoS) and it has access to MIH Information Server.
- **Step 2:** Mobile Node queries information about neighboring networks by sending the MIH_Get_Information request to Information Server. Information Server responds with MIH_Get_Information response. This information query may be attempted as soon as Mobile Node is first attached to the network.
- **Step 3:** Mobile Node triggers a mobile-initiated handover by sending MIH_MN_HO_Candidate_Query request to Serving PoS. This request queries information of potential candidate networks. It includes queries on QoS resources and/or IP address configuration method supported in the candidate networks.
- **Step 4:** Serving PoS queries the availability of resources at the candidate networks by sending MIH_N2N_HO_Query Resources request to one or multiple Candidate PoSs.
- **Step 5:** Candidate PoSs respond with MIH_N2N_HO_Query_Resources response and Serving PoS notifies the Mobile Node of the resulting resource availability at the candidate networks through MIH_MN_HO_Candidate_Query response.
- **Step 6:** Mobile Node decides the target of the handover and requests resource preparation by sending the MIH_MN_HO_Commit request to Serving PoS.
- **Step 7:** Serving PoS sends MIH_N2N_HO_Commit request to Target PoS to request resource preparation at the target network. Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit response.

- **Step 8:** When the resource is successfully prepared, Serving PoS sends MIH_MN_HO_Commit response to Mobile Node.
- **Step 9:** New layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network.
- **Step 10:** Mobile Node may send MIH_MN_HO_Complete request to Target PoS. Target PoS sends MIH_MN_HO_Complete request to previous Serving PoS to release resource which was allocated to Mobile Node. After identifying that resource is successfully released, Target PoS may send MIH_MN_HO_Complete response to Mobile Node.

To support service continuity the following extensions are proposed:

Step 3: Extension is needed in step 3 if the Mobile Node is looking for a particular set of services and the candidate networks must support them.

The primitive MIH_MN_HO_Candidate_Query request has to be modified to accommodate service information, which constitutes one of the criteria for identifying candidate networks. We propose to introduce an additional parameter called QueryServiceList as follows:

MIH_MN_HO_Candidate_Query.request (
DestinationIdentifier,
CurrentLinkIdentifier,
CandidateLinkList,
QueryResourceList,
IPConfigurationMethods,
DHCPServerAddress,
FAAddress,
AccessRouterAddress,

QueryServiceList,
)

The parameter QueryServiceList is the list of services that the Mobile Node is requesting. It has a type LIST(SERVICE) which contains 0 or more services. The data type SERVICE is defined as shown in TABLE 3.

Table 3 Definition of data type SERVICE

Data type name	Derived from	Definition
SERVICE	SEQUENCE(SERVICE_NAME, SERVICE_TYPE, SERVICE_PARENT_TYPE, SERVICE_EQUIVALENCE_ CLASS, SERVICE_KEWORDS, SERVICE_DESCRIPTION)	A type to represent a list of service in the access network.
SERVICE_NAME	OCTET_STRING	A type to represent a service name. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_TYPE	OCTET_STRING	A type to represent in which category the service belongs to. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_PARE NT_TYPE	OCTET_STRING	A type to represent ParentType of a service. A non-NULL terminated string whose length shall not

		exceed 253 octets.
SERVICE_EQUIVALENCE_CLASS	OCTET_STRING	A type to represent equivalence services (any EquivalenceClass of the service defined by the Service Provider; it can also be service with the same ParentType). The value is a non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_KEYWORDS	OCTET_STRING	A type to represent keywords of a service. The value is a non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_DESCRIPTION	OCTET_STRING	A type to represent the description of a service. A non-NULL terminated string whose length shall not exceed 253 octets.

SERVICE_NAME identifies a service instance. A service instance has a unique SERVICE_NAME but a SERVICE_NAME may be given to several service instances. In a future mobile environment where anybody can be a service provider and a service can be anything a SERVICE_NAME will not be standardized and the same name can be used by multiple service providers to denote multiple different services [46].

SERVICE_TYPE identifies the type of the service. As for SERVICE_NAME the SERVICE_TYPE can be ambiguous in a future mobile environment.

SERVICE_PARENT_TYPE identifies parent service type that the current service type is derived from. The SERVICE_PARENT_TYPE is necessary in the service matching when a partial match, i.e. only a subset of features is required [49].

SERVICE_EQUIVALENCE_CLASS identifies the equivalence class that the service type belongs to. This identifier is necessary to enable the usage of multiple languages to denote a service.

SERVICE_KEYWORDS contains attributes that help narrow the scope of the service matching.

The SERVICE_DESCRIPTION contains information needed to use the service.

Step 4: It is necessary to modify the primitive MIH_N2N_HO_Query_Resources request to accommodate service information. The researcher proposed to introduce an additional parameter called ServiceRequirements as follows:

```
MIH_N2N_HO_Candidate_Query.request (
                DestinationIdentifier,
                QoSResourceRequirements,
                IPConfigurationMethods,
                DHCPServerAddress,
                FAAddress,
                AccessRouterAddress,
                CandidateLinkList,
                ServiceRequirements,
                )
```

The parameter ServiceRequirements is the service requirements that the Mobile Node is requesting. It has a type LIST(SERVICE) which contains 0 or more services.

Step 5: In this step the list of candidate networks will be returned by Candidate PoS with the primitive MIH_N2N_HO_Query_Resources response and forwarded to the Mobile

Node through the primitive MIH_MN_HO_Candidate_Query response. No modification is required for these two primitives but in order to compile the candidate list the Candidate PoSs will have to carry out a service discovery and matching in addition to the standard resource query.

Step 6: To select the target network for handover, the Mobile Node must choose the most appropriate available services and requests resource preparation by sending the MIH_MN_HO_Commit request to Serving PoS. In addition to resource preparation, appropriate actions must be carried out on the network side to perform the service transfer to replace current service instance with new equivalent service instance.

The primitive MIH_MN_HO_Commit request must be extended with an additional parameter TargetServiceInfo as follows:

MIH_MN_HO_Commit.request (
DestinationIdentifier,
LinkType,
TargetNetworkInfo,
TargetServiceInfo,
)

The parameter TargetServiceInfo has LIST(SERVICE) as type and contains the names of services to be prepared for the handover.

Step 7: Serving PoS sends MIH_N2N_HO_Commit request to Target PoS to request resource preparation at the target network.

The primitive has to be extended with an additional parameter RequestedServiceSet of LIST(SERVICE) type which indicates the services that needs to be prepared for handover as follows:

MIH_N2N_HO_Commit.request (
DestinationIdentifier,
MNIdentifier,
TargetMNLinkIdentifier,
TargetPoA,
RequestedResourceSet,
RequestedServiceSet,
)

Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit response, which is extended with the parameter AssignedServiceSet of LIST(SERVICE) type as follows:

MIH_N2N_HO_Commit.response (
DestinationIdentifier,
Status,
MNIdentifier,
TargetLinkIdentifier,
AssignedResourceSet,
AssignedServiceSet,
)

Step 8: When the resource is successfully prepared, Serving PoS sends MIH_MN_HO_Commit response to Mobile Node. The primitive MIH_MN_HO_Commit.response is also extended with an additional parameter called TargetServiceInfo of LIST(SERVICE) type as follows:

MIH_MN_HO_Commit.response (
DestinationIdentifier,
Status,
LinkType,
TargetServiceInfo,
)

Step 9: New layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network. In addition to these standard procedures for the MIH, we propose to perform also service transfer procedures to ensure service continuity.

Step 10: Mobile Node may send MIH_MN_HO_Complete request to Target PoS. Target PoS sends MIH_N2N_HO_Complete request to previous Serving PoS to release resource which was allocated to Mobile Node. After identifying that resource is successfully released, Target PoS may send MIH_MN_HO_Complete response to Mobile Node. In addition to the release of radio resources, it may also be necessary to release service resources.

To enable service continuity, the Media Independent Command Service (MICS) must accommodate the modified commands as described above.

D. Proposed extension to the MIIS

The Media Independent Information Service (MIIS) provides a framework and corresponding mechanisms by which an MIHF entity can discover and obtain network information existing within a geographical area to facilitate handovers. To ensure service continuity, it is also necessary with extensions in the MIIS as proposed in Paper 1 [50]. More specifically, new Information Elements (IEs) containing service information must be added to the standard set as follows:

- IE_NET_SERVICES is introduced in the Access Network Container which contains information about services supported by the access network. This element is used when the network is offering the same services at every PoAs. If the network does provide different services at different PoAs, this information element must be left empty. When encountering an empty IE_NET_SERVICES the service discovery will have to proceed to the respective PoA container to carry out the service matching.
- IE_POA_SERVICES is introduced in the PoA Container which contains information on services supported at the PoA. For networks offering the same services at all the PoAs, this element must be empty since the information about the services is already specified at the network level.

Figure 21 depicts the example on how the new service information is included in the Information Elements.

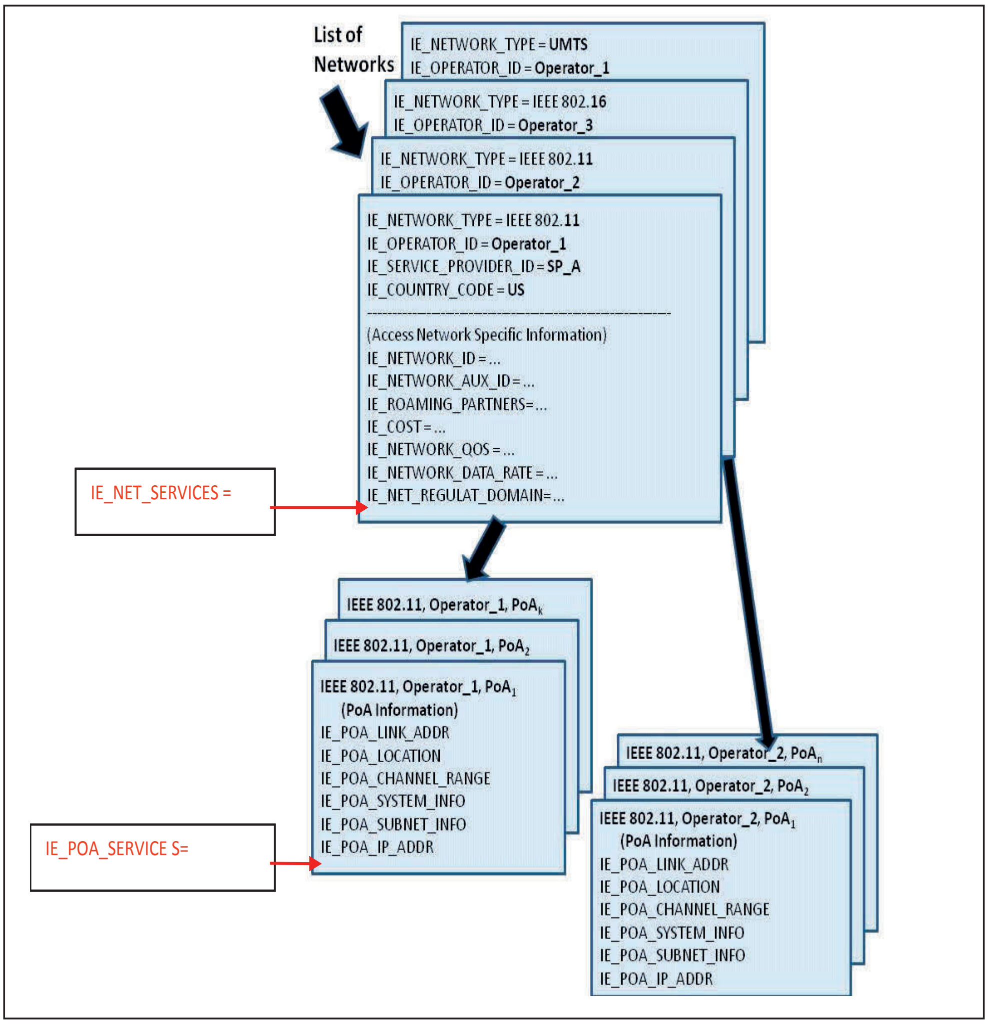

Figure 21 Information elements with service information

3.3 Service discovery

The work described in this section is covered by paper 2, 3, 4, 5, 6, 9 and 10.

One of the fundamental elements necessary for future mobile services is the service discovery. Indeed, with the plurality of networks offering a large variety of services mobile devices need to have a sound and efficient service discovery, which enables fast discovery and consumption of services. In this section, service discovery intended for future mobile environment is proposed.

3.3.1. Requirements on the future service discovery

From the scenario of future mobile environment where the mobile device can be connected simultaneously to several network systems offering heterogeneous sets of services using different names, the requirements on the future service discovery are deduced and analysed. The problems and conflicting needs are also identified.

1. A service in future ubiquitous communication system can be anything. This leads to a large variety of names, which makes the limitations in number of digits or in types of characters quite difficult or almost impossible. The consequence is the situation where the same name or word can have several meanings and denote different services. Ambiguity and confusion are hence introduced. Thus, *the service discovery must be capable of handling different services with same names without confusion.*
2. A service can be introduced by anybody at any time. The result is that not all services can be standardized as in current service discovery systems where a service is well specified and has a uniquely defined name. Another consequence is that the same service can be given different names by different service providers. Since there is no regulation about a service definition, a service may be close to another one but not 100% similar. There are two cases as follows:

- Service A contains similar feature elements with Service B but has also different elements as shown in Figure 22 - 1). The intersection of A and B is not empty and different from both A and B.

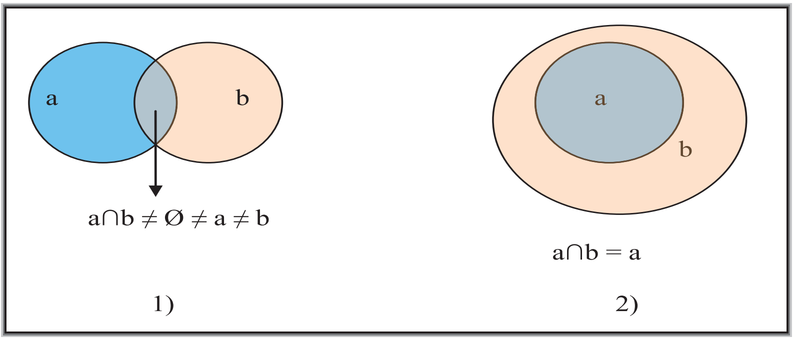

Figure 22 Relations between services

- Service A is a subset of Service B as shown in Figure 22 – 2) since all the elements of A are also elements of B.

A service can have several names and they can be in other languages than English, e.g. Norwegian, Chinese, Urdu, etc. Hence, *the service discovery must be capable of services with multiple names in multiple languages and must allow the introduction of any service anytime by anybody.*

3. In a mobile environment, services must be discovered very fast which means that the service discovery must be very efficient. Hence, this calls for *shorter service name*.

4. It is crucial not to misunderstand about a service or to confuse one service with another one. Hence, *the service discovery must be error-free*. This calls for longer service name, which is in conflict with requirement 3.

5. All the services available in an area must be discovered. Hence, *the service discovery must be sound.*

6. It is also essential to be able to verify that a service is offering what it announces. Thus, it must be possible to extend the service discovery with *verification functions*.

7. It is also essential to be able to conclude that a service is trustful. Therefore, it must be possible to extend the service discovery with *security functions to validate a service*.
8. The user must be able to move everywhere in the world. Thus, the service discovery must be capable of *functioning ubiquitously*.
9. Since there are currently many services and service discovery systems, it is important to ensure interoperability. Hence, the service discovery must be *capable of discovering current existing services and operating with existing service discovery systems.*

More details about future service discovery requirements can be found in Paper 3 [46].

3.3.2. Service naming

One of major components of the service discovery is the service naming, which has to be sound.

In the near future, multiple connections can be established simultaneously and the mobile device will have access to a wider range of services. The challenge for the mobile device is then to discover the services, figure out their functions, select the desired services, acquire their protocols, syntax and formats and finally invoke the desired services. In other words, the mobile device must know what a service with the name "A" is offering. This could be quite difficult and time consuming when multiple heterogeneous network systems with different concepts and conventions are involved. In fact, the process can be indeterministic and unable to accomplish. It is hence crucial to have a sound service naming scheme.

A. Service naming policy

The high-level requirements on the service naming schemes are as follows:

1. The service naming must allow fast service discovery: when the mobile device moves from one area to another one, it should be offered the possibility to discover new

services quite quickly in order to replace the service currently in use before they disappear.

2. A service name shall uniquely identify a service: once a name is registered for a service in a particular domain, there will not be any similar name to that service unless if it is under other administrative domain.
3. A service name shall be comprehensive and give an indication about the services' purposes and functions: from the service name anybody should be able to guess what the service is offering.
4. A service name shall be sufficiently simple to allow a fast service matching: the service matching here means the result of comparisons between the user requirements and the service availability, which occurs after the services are discovered. So, to offer service continuity to a mobile user who is in a changing environment it is crucial to have fast service matching which again requires simple service names.
5. It shall be possible to assign several names to a service in different languages: in a global environment with multiple cultures and languages, it is necessary to support service names in different languages.
6. It shall be possible to introduce aliases for a service: the reason for having aliases is to provide for transparency [51]. By having aliases it will help to identify the service function and makes the name easier to promote and remember. It also facilitates transfer of the service to different infrastructure whenever required.
7. It shall be possible to use a short name for a service: as the service naming is used to uniquely identify and addressed the service, hence a long name might be confusing and error prone. So it is possible to use a short name for a service. As an example explained in Unmanaged Internet Architecture (UIA) [52], the name work-pc can be used instead of using long globally unique names like work-pc.nor.item.ntnu.no as was used in DNS. However, in order to do that, the name should be identical to others in its own administrative domain.
8. It shall be possible and simple to introduce new services: as the services are uncountable in numbers and there can be new services introduced every day, the

services need hence to learn about each other dynamically and cooperatively so the service naming should allow for introducing new services and it should be simple.

9. The service name scheme should be distributed and should allow more service providers to introduce any new service: there will always be multiple service providers who introduce new services in an uncoordinated way and the service name scheme should be able to cope with this.
10. The service naming scheme should be flexible and interoperable with existing service naming schemes: since there will always exist other service naming, it is desirable that the new service naming can interoperate with them.

B. Service naming for future mobile services

The proposed service naming scheme must fulfill the requirements specified above as follows:

1. The service naming must allow fast service discovery:
   - This means that the service name must be as short as possible. We propose to use the commonly known noun of a service as its name. For example, the service to find and book a restaurant is simply named “restaurant”.
2. A service name shall uniquely identify a service:
   - In order to avoid name collision, each service name should uniquely identify a service. Unfortunately, a short name as specified in the previous bullet may not be sufficient to uniquely identify a service because a noun may have different meaning in different context. For example, the word pen has five meanings:

     - Pen – a writing implement with a point from which ink flows
     - Pen – an enclosure for confining livestock
     - Playpen, pen – a portable enclosure in which babies may be left to play
     - Penitentiary, pen – a correctional institution for those convicted of major crimes
     - Pen – female swan

- Consequently, it must be possible to expand the short name to a more composite or complete name, which uniquely identifies the service. It is necessary to have hierarchical structure, which allows gradual navigation towards higher hierarchical level and stepwise narrowing of the identification of the service and removal of ambiguity.
- For example, if "pen" is not sufficient to uniquely identify the service, a higher level component name may be fetched and use in the service matching. For example, the higher level name component of pen which is "writing" can be used in the service matching. If it is still ambiguous, the higher name component, namely "instrument" can also be fetched and used.

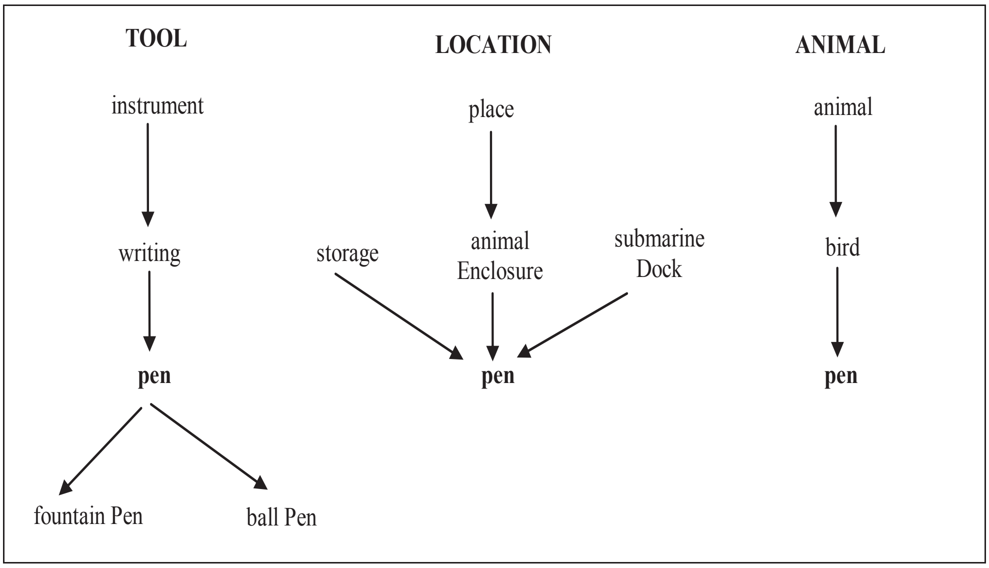

Figure 23 Example of the name structure hierarchy

- The researcher proposed that a service name structure which is inspired of the Uniform Resource Identifier (URI) as follows:

  - Semantically, the full name of a service will reflect the classification of the service in certain ontology. Figure 23 shows that the service "pen" may be belong to three different ontologies.

- Syntactically, a name consists of several component names separated by period as follows:

  - <componentnameN>..<componentname1>
  - E.g.: instrument.writing.pen

  The shortest name is always used first. If it is ambiguous, the service name will be gradually expanded until the service is uniquely identified.

- To name a service properly, it is hence necessary to associate a service to ontology and to determine its upper class.

3. A service name shall be comprehensive and give an indication about the service's purposes and functions:
   - Since almost everything can be a service from restaurant, cinema, hotel, taxi, etc. The researcher proposed to use the most common and shortest noun as name for a service. For example, there are many words which all denote a restaurant: restaurant, bistro, cafeteria, brasserie, etc. The most common word which is "restaurant" will then be used.

4. A service name shall be sufficiently simple to allow a fast service matching:
   - To allow fast service matching, the service name should not be too long but it must neither be too short and hence be confounded with other services. Such confusion will require a second round of name comparison which will be time consuming. With the proposed service name scheme a service will have composite name that can be quite long. However, it may not be necessary to use the full name but only a partial name like "writing.pen". It is a challenge to choose the optimal length for service names. The researcher proposed an empirical approach for the determination of the service name length in which the shortest name of a service with only one component name should be used first. When a need for further expansion in the

service matching occurs the service name will update with a longer name with the number of component names corresponding with the expansion required in the service matching.

5. It shall be possible to assign several names to a service in different languages:
   - A service name may have different names in different languages. For example, the service "taxi" is called "teksi" in malay, "taksiliito" in finnish, "cab" in uk, etc. These different names should belong to an equivalent class called taxi as follows:

     [taxi] = {x ϵ S| X ~ taxi} S: set of services

     ~ is an equivalent relation which is

     - Reflexive: a ~ a
     - Symmetric: if a ~ b then b ~ a
     - Transitive: if a ~ b and b ~ c then a ~ c

     Alternative, the equivalent class taxi can be expressed as follows:
     [taxi] = {taxi, teksi, taksiliito, cab,..}

6. It shall be possible to introduce aliases for a service:
   - An alias is just another name that can be added to the equivalent class of the service as specified earlier.

7. It shall be possible and simple to introduce new services and multiple service providers should be able to introduce new services:
   - To introduce a new service, a new name has to be created. To avoid name collision which can create confusion and increase the service matching time, it is necessary to check whether a name has been used before. If it is the case, it must be possible to investigate whether it is the same service or not. If the services are similar there is no need to introduce a new one. If the services are different it may be necessary to

examine the differences. If the differences are minor and the services are related to each other, it is necessary to classify the new service as either parent or child of the existing service class. If there are too many differences, a new service name should be chosen for the new service. One approach that can be used is by implementing a domain specific ontology, where the same term may be used to denote different concepts. It is able to determine the relevance of an information source before accessing the underlying data and supporting wider accessibility of data via multiple ontologies and interoperation across them based on semantic relationships such as synonyms, hyponyms and hypernyms [53]. For name checking it is necessary to have a service registry which is ubiquitously available. For each service there should also be description or references to description that enable service providers to understand about the services. The service registry must also offer registration and deregistration service.

8. The service naming scheme should be flexible and interoperable with existing service naming schemes:
    - By allowing alias the proposed service naming scheme will allow associating a service class with existing services defined by existing systems such as Jini, UPnP, OMG, etc.

3.3.3. Design of service discovery and service matching

As stated earlier, in the near future, a service offered to the user can be anything and the service discovery including the service naming will meet the challenge of coping with this dynamicity. Moreover, in a multi service provider environment, services with the same name can be partially or totally different while services with different names may be similar.

To cope with this dynamicity and flexibility, there is a need for a very open and extensible service naming scheme which allows a very flexible definition of the service name at the same time as it supports an efficient, reliable and sound service matching.

The design of the future service discovery architecture is also presented in Paper 3. In order to have a more precise and formal view on the whole process of the service discovery and service matching, the Unified Modeling Language (UML) [17] is used:

As shown in Figure 24, 4 use cases are identified which are requested by two players, the user and the service provider as follows:

- User:
  - Discover the services
  - Select a service
  - Find a service
- Service provider:
  - Register a service

The four use cases represent the high level requirements of the future service discovery system.

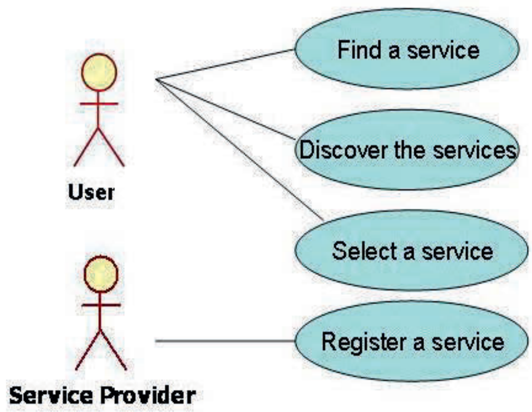

Figure 24 Use case diagram of future service discovery system

A. Use case Register a service

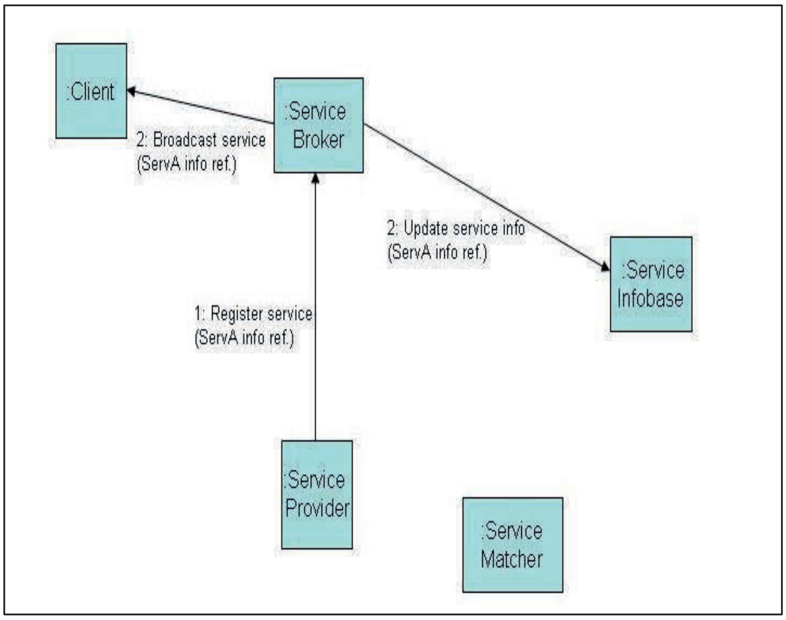

Figure 25 Collaboration diagram for Register a service

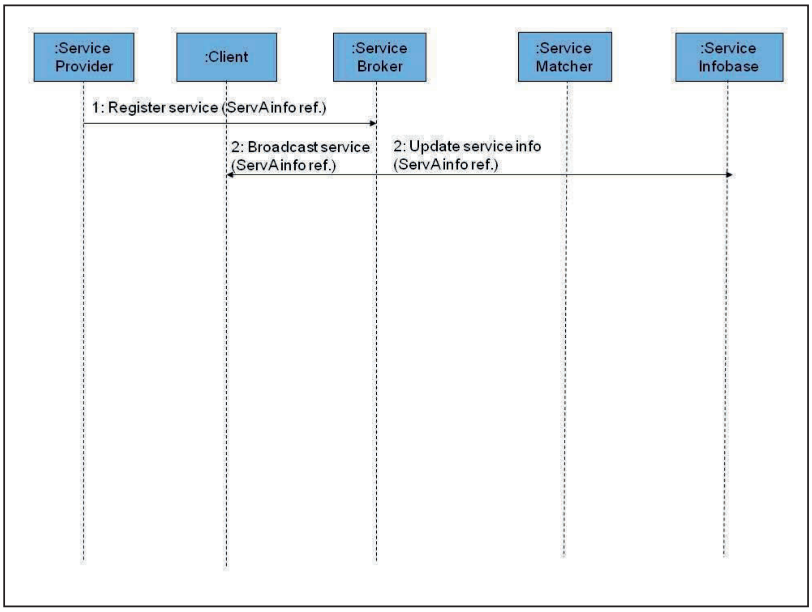

Figure 26 Sequence diagram for Register a service

In the current situation where the mobile is connected to a mobile network (e.g. GSM or UMTS), the service that is offered by the network has a standardized naming and service number, for example service name telephony with a dedicated service number which uniquely identify the service. From there if the client wants to find the service, a direct syntax matching can be done but in the future, the mobile will be connected in a heterogeneous network thus allowing the mobile to be connected to other services as well and with different names that the client might not know. The issue is how the device can find out about the services and how the service is named. The syntax matching can no longer rely on the service name, service number, service type and attribute. To solve this issue, the researcher proposes to come out with semantic service matching based on ontology. The convention on how a service name should be is clearly specified in the Paper 2 [54]. A template to be used by a Service Provider at registration of services is illustrated in Table 4.

Table 4 Example of template for future service naming scheme

Field	Description
Service name	A short 'noun' and common service name is used for e.g. "restaurant" to refer to place that serves food.
Service equivalent class	• It is used for a service with different names but having same service instances. • It can also refer to services called by different languages (alias). • For e.g. [restaurant] = {restaurant, ristorante, cafe, warung, …}
Hierarchical structure	The hierarchical structure of the service name shows in which classification the service is in. This is very important especially for the services with the same name that can be different (e.g. Book that refers to an object or book which refers to a verb that means reservation)
Next level service	• If the service name is not sufficient during service matching, next level service name will be checked until there is no ambiguity and the final service definition is retrieved.

	• It can also be used to find partial match (e.g. If the taxi service cannot be found, the service matching will go to the next level to see in which group/classification the service is in and suggest alternatives, in this case the bus or train which falls under the transportation group/classification is found as partial match)
Service name structure	The service name is in the uri format which contains <componentnamen>…<componentname1> For e.g. Reservation.place.book

This template will be written in XML and contains in the service Infobase for Service Provider guidelines. Figure 25 shows a collaboration diagram for a use case "Register a service" and Figure 26 depicts the sequence diagram for the use case showing the sequence of execution. As shown in Figure 25 the use case starts as follows:

1. When the Service Provider wants to register a service, it first needs to ask the Service Broker to check on the existing service name. If the service already exists, the Service Provider will register the service with the same service name otherwise the Service Provider will need to look at the service equivalent to check on the same service with a different name. The Service Provider registers the service with service information reference (template shows in Table 4).
2. Once the Service Broker receives the registration, it will broadcast the service with service information reference to the Client and at the same time update the service information to the Service Infobase.

B. Use case Discover the services

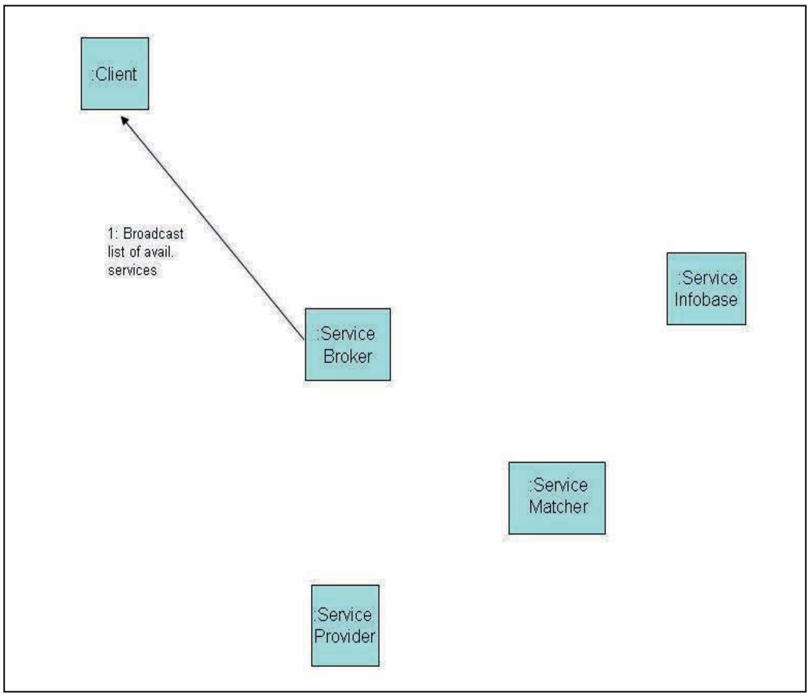

Figure 27 Collaboration diagram for Discover the services – alternative 1

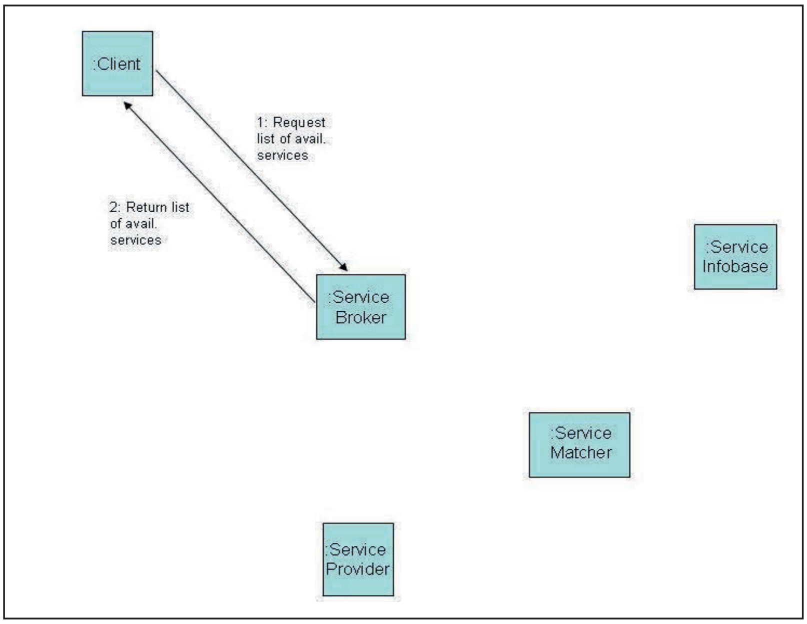

Figure 28 Collaboration diagram for Discover the services – alternative 2

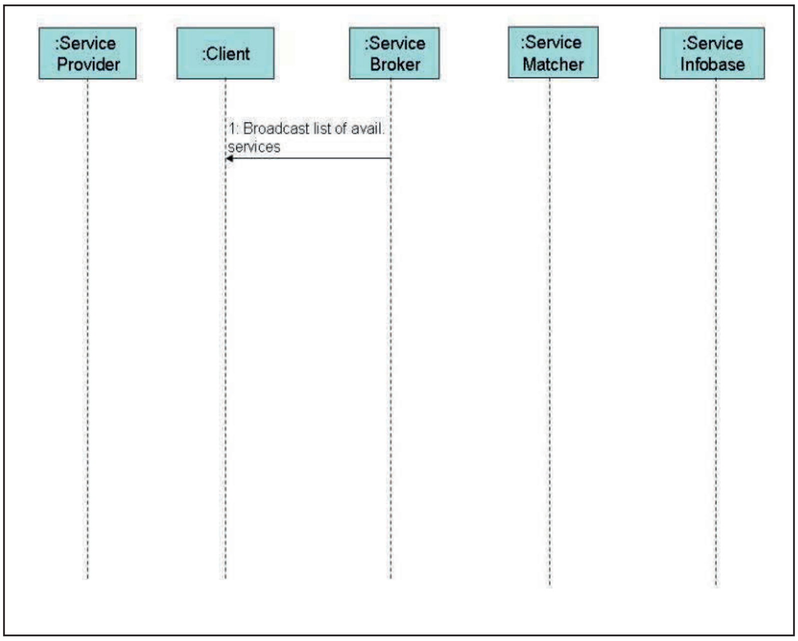

Figure 29 Sequence diagram for Discover the services – alternative 1

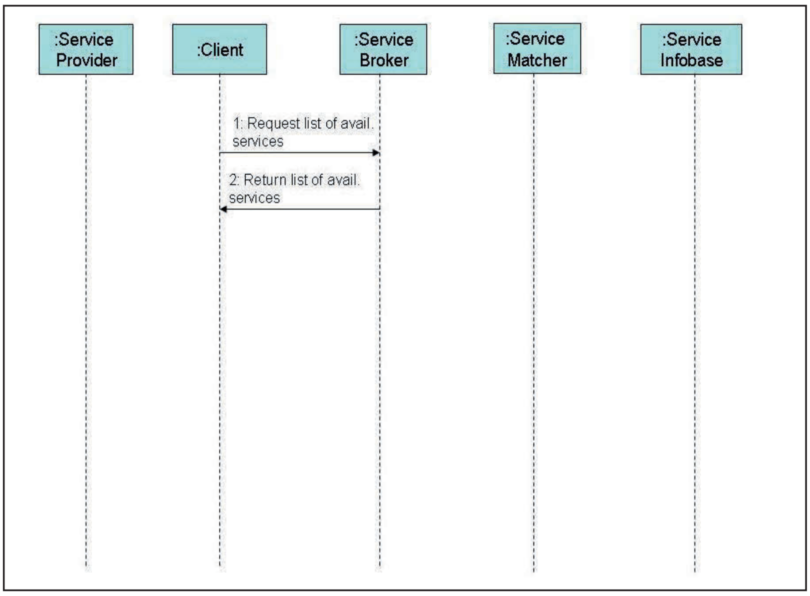

Figure 30 Sequence diagram for Discover the services – alternative 2

Figure 27 and Figure 28 show collaboration diagrams for a use case discover the services and Figure 29 and Figure 30 are the sequence diagrams for the use case with two alternatives, whether the service broker is broadcasting the list of available services or the client requests for the available services to the service broker. As shown in Figure 27 the use case starts as follows:

1. The Service Broker broadcasts the list of available services to the Client. There are two ways on broadcasting the services, either by periodically broadcasting the services within a certain amount of time or upon new arrival/registration of a service. The broadcast method requires less power from the mobile phone since it can simply accept the list of available services from the service broker. However, the Client will need to synchronise before being able to get the list of available services and this may take time. The consequence is that the mobile device may loose contact with former service components before the new one is discovered and result onto service discontinuity.

Figure 28 is the alternative for the use case discover the services that start as follows:

1. The Client first initiates a request on the list of available services. The request for the available services could be done in two ways: either the Client subscribes for the service or the Client just moves to a new area and would like to enquire about the available services.

2. Once the Service Broker receives the request, it will return the list of available services. For this method, the requester (Client) and the Service Brokers (there could be several service brokers at a location) are supposed to listen and respond to the request or in the case of Media Independent Handover (MIH), there is a dedicated channel (or address) reserved for a unique and centralised Service Broker and the Client can send the request directly to it.

Therefore some important points need to be considered such as:

- The list of available services must be usable and comprehensive but not too exhaustive due to the limitation in bandwidth of the wireless link, storage capacity of the mobile device, high latency, etc.
- The list should not contain all the service details but only the strict minimum such as service name and reference to service information.
- The list must have syntax and semantics which are comprehensive for both the Client and the Service Broker.

C. Use case Select a service

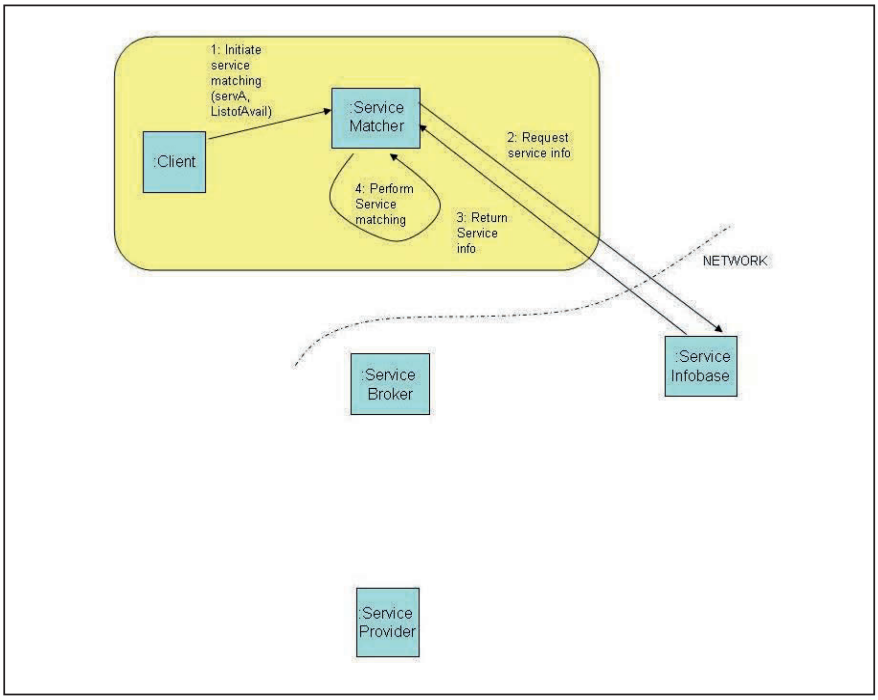

Figure 31 Collaboration diagram for Select a service – alternative 1

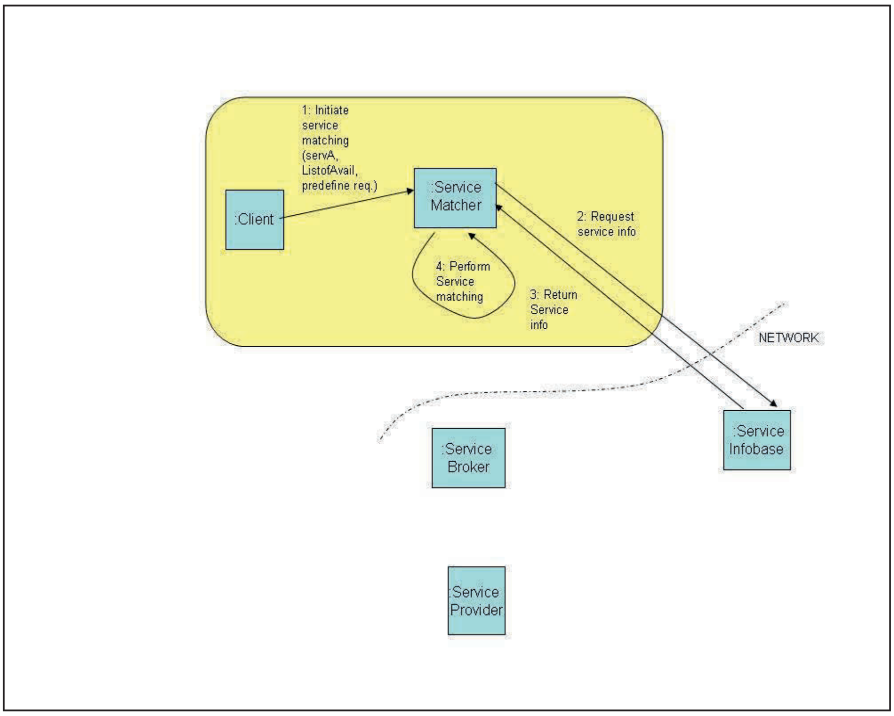

Figure 32 Collaboration diagram for Select a service – alternative 2

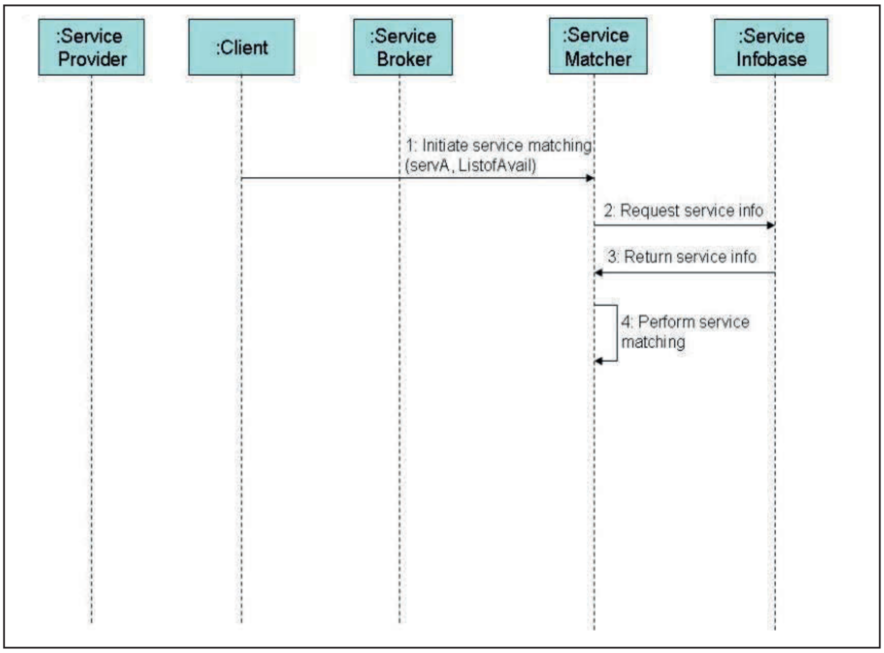

Figure 33 Sequence diagram for Select a service – alternative 1

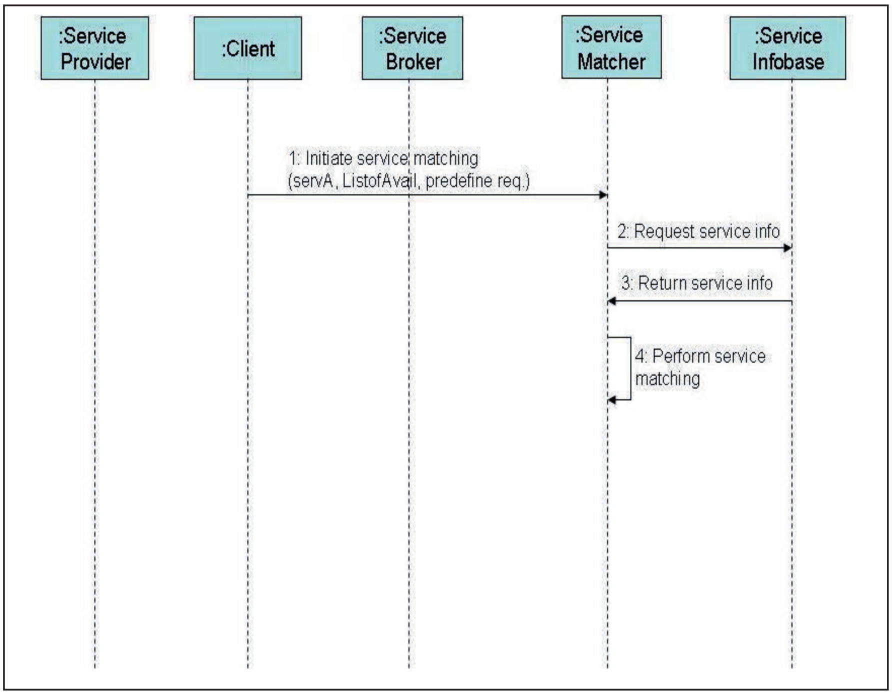

Figure 34 Sequence diagram for Select a service – alternative 2

Figure 31 and Figure 32 show the collaboration diagrams for the use case “Select a service” and Figure 33 and Figure 34 depict the sequence diagrams for the use case with two alternatives: either to have the selection of the service manually by the client or to automate the selection. As shown in Figure 31 the use case select a service (manually) starts as follows:

1. The Client initiates service matching by sending the service name and the list of available services to the Service Matcher.
2. The Service Matcher then sends a request of service information to the Service Infobase.
3. The Service Infobase will return the service information to the Service Matcher.
4. Upon receiving the service information, the Service Matcher will perform service matching.

On the other hand, in the second alternative as illustrated in Figure 32, the selection of the service is done automatically and the use case starts as follows:

1. The Client initiates service matching by sending the service name, the list of available services together with predefine requirements (such as user personalisation and device adaptability) to the Service Matcher.
2. The Service Matcher then sends a request of service information to the Service Infobase.
3. The Service Infobase will return the service information to the Service Matcher.
4. After the service information is received, the Service Matcher will perform service matching and check also the predefined requirements sent by the Client. By implementing this method, optimization will be achieved and results in an efficient service matching because in some cases there are possibilities of the matching result that does not comply/suit with user personalisation or user-device adaptability and is hence a waste of time.

There are some important points that need to be considered as follows:

- It is necessary to introduce metrics for defining the ‘best offer’ for an automate service matching or via optimization algorithm implemented on the client’s side, such as shortest routing path or lowest hop count.
- The Service Matcher could be located either in the Client side or in the Network side. The advantage of having the Service Matcher in the Client side is to ensure the mobile phone’s full control on the service matching but this will consume more power. Having a service matching function in the Network side will reduce the processing time of the client and hence increase the performance of efficiency but there is a risk of connection loss.
- The service information in the Service Infobase can be in the World Wide Web (WWW) or stored in the MIH. If the Service Infobase is in WWW it will be

accessible by any service providers. Being integrated in the MIH will simplify the implementation.

D. Use case Find a service

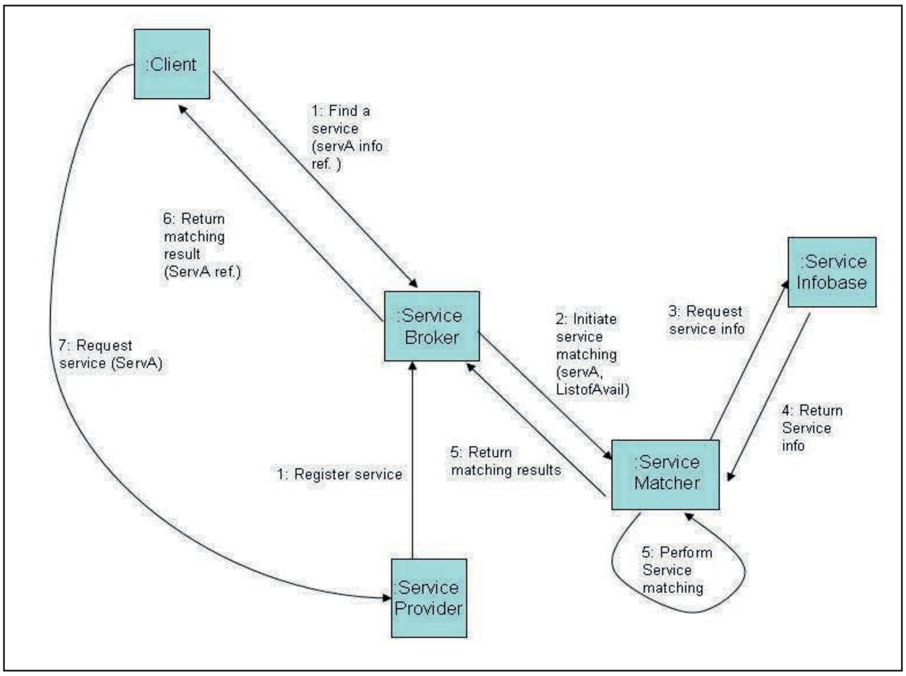

Figure 35 Collaboration diagram for Find a service

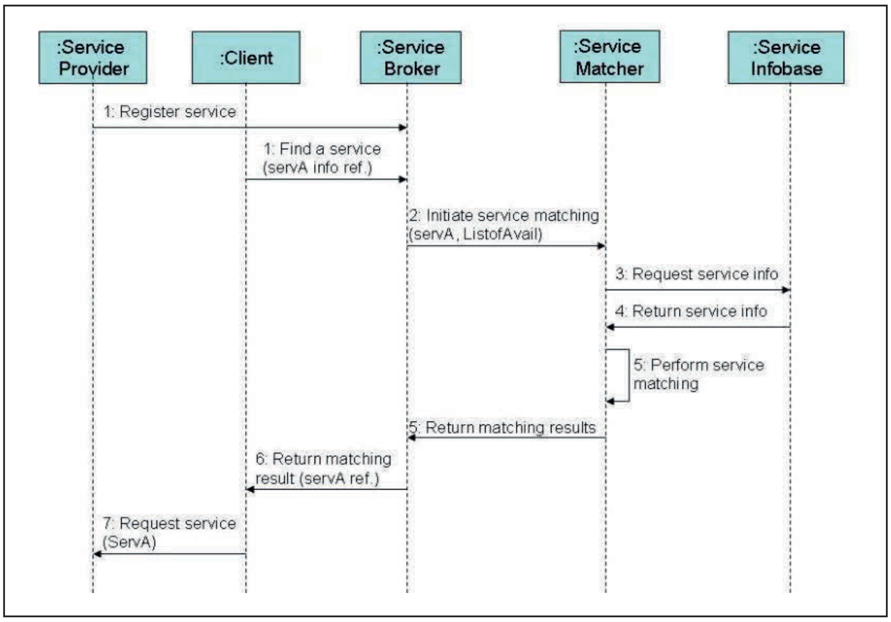

Figure 36 Sequence diagram for Find a service

Figure 35 shows collaboration diagram for a use case “Find a service” and Figure 36 depicts the sequence diagram. As shown in Figure 35 the use case starts as follows:

1. The Service Provider registers for a new service and when a Client moves to a location, it will find a service by sending service information reference to the Service Broker.
2. Service Broker will initiate service matching by sending the service name and the list of the available services to the Service Matcher. In this case, the Service Matcher is located at the network side.
3. The Service Matcher then sends request of service information to the Service Infobase.
4. The Service Infobase will return the service information to the Service Matcher.
5. After receiving the service information, the Service Matcher will perform service matching and return the matching results to the Service Broker.

6. The Service Broker then will return the matching result together with service reference to the Client.
7. Upon receiving the service matched, the Client will send request for the service directly to the Service Provider.

3.3.4. Implementation and validation of the proposed service discovery

To verify the proposed service discovery an implementation and validation has been carried out (See Paper 5) [18]. The main contribution is the introduction of three main attributes using the OWL Lite for semantic matching during the service discovery which are:

- `EquivalenceClass`
- `ParentType`
- `Keyword`

As in the future mobile environments, a service can be anything and introduced by anybody. Hence, same or equivalent services may have different names and services with the same name or type may be completely different. This poses challenges and the existing service discovery system is not capable of handling these situations. The service name and service type should not be restricted to any length or format. The service provider must have the freedom to the name and type as pleased. This requirement poses problems for the service matching to succeed.

More semantics are introduced in the service description. In addition to regular parameters like Name, Type, State Variable and Actions which provide the syntaxes of the service additional parameters like Keywords, ParentType, EquivalenceClass are introduced to provide more semantics to the service. They will be used in the service matching to find the requested services in an efficient and unambiguous manner. Their usage will be elucidated by the use scenarios in later section.

A. Overview of the system architecture

Figure 37 illustrates the system architecture for future service discovery.

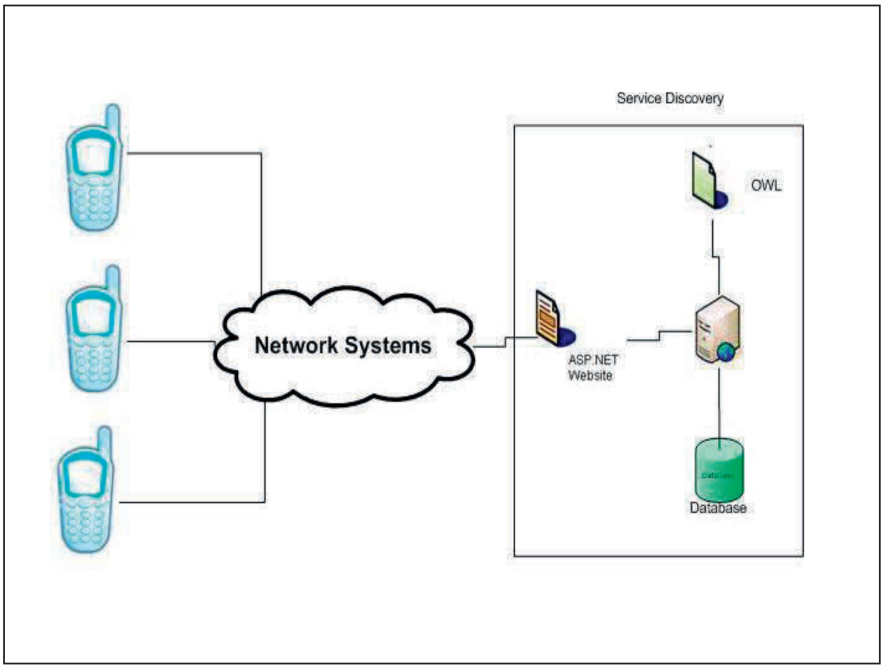

Figure 37 Future service discovery system architecture

The major components of the architecture are:

1) Web Service Client – This component is an application running on the remote mobile terminal. The Client on behalf of the user can ask/search for a service. It can also, on behalf of the Service Provider introduce and register a new service. The protocol used between the Client and the actual XML Web Service is Simple Object Access Protocol (SOAP) [55].

2) Components of the service discovery system consists of:

   i) ASP .Net [56] Website (XML Web Service) – This service acts as an interface between the Client/Service Provider and the System Server. The IIS Internet Information Services 5.01 or later is customized in order to publish the ASP.Net pages on the Internet.
   ii) Database – This component acts as a service repository which stores the services registered by the service provider.
   iii) OWL file – is used for semantic matching with the OWL Lite (EquivalenceClass and ParentType).

B. Service discovery testbed

The hardware as well as the software components used for the system implementation is described below:

1) Hardware - A computer with installed Windows XP is used as a server. The server can process several requests from multiple clients in parallel. The Client can request the server by using any platform or browser to retrieve the desired services.
2) Software

   Operating System

   - Server side: Since the application is developed using .Net technology, the .Net framework version 2.0 is installed on the server. Microsoft Windows based operating system is the minimum requirement for this version of .Net framework. Therefore Microsoft Windows XP was installed on the server.
   - Client: The Client is independent of any architecture and hence can access the services by using any industry standard Internet browser from any operating system that has access to the Internet.

Programming languages: For the implementation of this project Microsoft Visual Studio 2005 IDE for C# was used. The application was developed in C# whereas the website was developed using ASP.Net technology.

- C#: C# was mainly used as an implementation language because of its convenient programming capabilities, the object oriented paradigm it supports, type-safety and wide range of libraries available as a part of .Net framework [57]. Another reason to use C# was the OwlDotNetApi that was mainly used to access OWL file in order to define parent as well as equivalent classes for service types. The detail of this API can be found on [58].
- ASP.Net: ASP.Net is used for the XML web service and the web form interface for accessing the web service. Reason for using ASP.Net:
  - for building XML web service because it makes exposing and calling web services very simple.
  - for Client website because of the flexibility of interface it provides and support for the mobile devices. By building a website in ASP.Net one has to write the code once and the ASP.Net automatically generates pages based on the device they are called.

Support for ontologies: As the semantic meaning in this project of service discovery is being achieved by ontologies they were continuously monitored manually by using Protégé 4.0.2 [59]. OWL Lite is chosen in the implementation of this project due to its simplicity and the dynamic nature of the project even though it is known that it has some limitations as compared with the OWL DL and OWL Full. Since the services can be introduced by anybody and there is no control mechanism on introducing the services, therefore it is required to avoid complexity in the system usage and for that only the EquivalenceClass and the ParentType properties is used which can be accomplished by the OWL Lite.

C. Sequence Diagram

The call sequence diagrams are used to explain the flow of the four main methods in the future service discovery.

▪ ***Registration a service***

This method is used for the Service Provider to register a new service as illustrated in Figure 38.

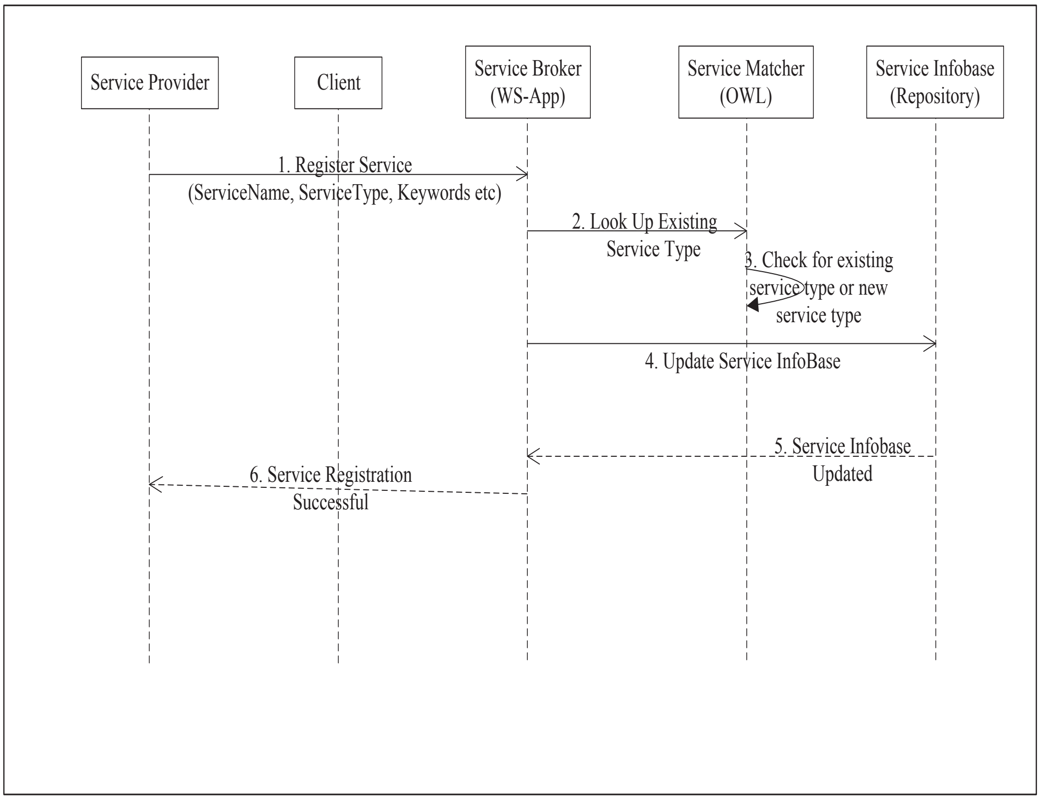

Figure 38 Call sequence diagram for Registration a service

i) When a Service Provider wants to register a service, he/she will first type the particular details of the service (e.g. the Service Name, Service Type and Keyword).

ii) The Service Broker which is the Web Service Application will then lookup for Equivalence Service Type.

iii) The Service Matcher which in this case is the OWL Lite will check for existing Service Type or new Service Type.

iv) The Service Broker will then update the Service Infobase which is the Service Repository of this system.

v) After that the Service Infobase is updated.

vi) The service registration is successful and Service Provider can see the new service registered in the service list.

- ***Discovering services***

This method is used for the Client which wants to find all available services as illustrated in Figure 39.

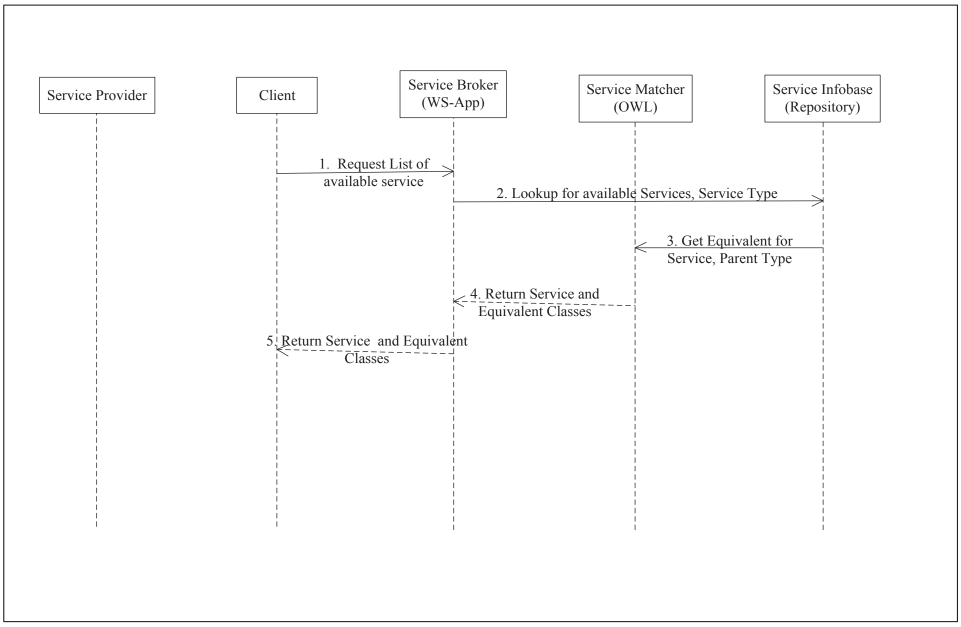

Figure 39 Call sequence diagram for Discovering services

i) When the Client wants to discover available services, it may request for list of services to the Service Broker.

ii) The Service Broker then will lookup for available services from the Service Infobase.

iii) The Service Infobase will then get the Equivalent for Service and Parent Type from the Service Matcher (OWL).

iv) The Service Matcher will return the available services and all Equivalent Classes to the Service Broker.

v) The Service Broker will return the services available and its Equivalent Classes to the Client.

- ***Service Request***

This method is used if the Client wants to find a particular service or set of services as illustrated in Figure 40.

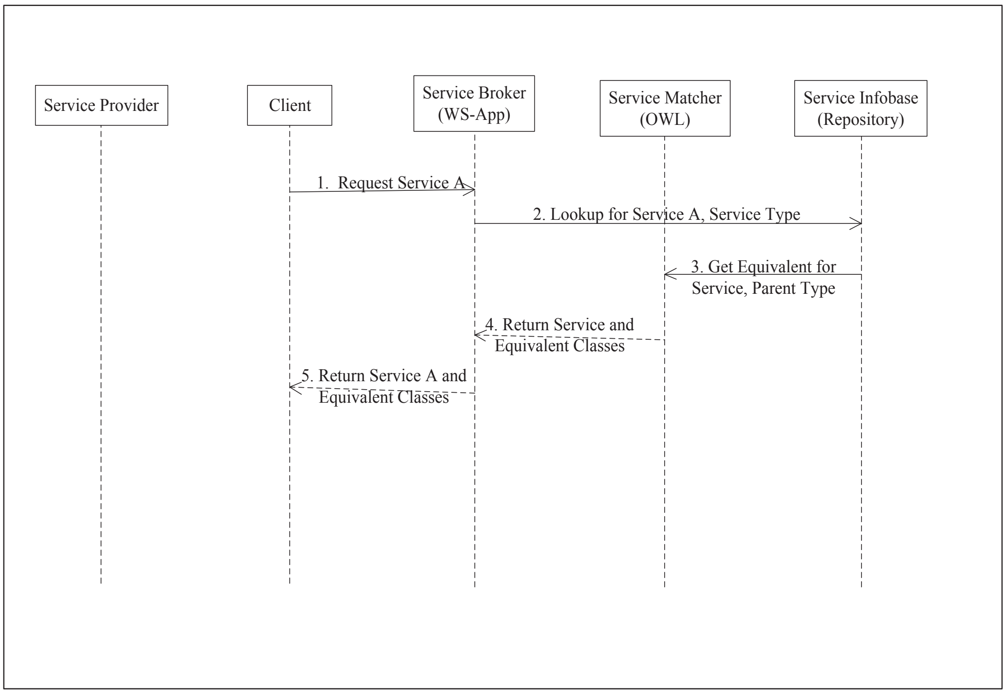

Figure 40 Call sequence diagram for Service request

i) When the Client wants to request for a particular service, he may find the service either by using the Service Name or the Service Type.

ii) The Service Broker will lookup for a particular service A and its Service Type in the Service Infobase.

iii) The Service Infobase will then get the Equivalent for Service A and its Parent Type from the Service Matcher (OWL).

iv) The Service Matcher will return the requested Service A and its Equivalent Classes to the Service Broker.

v) The Service Broker will return the Service A and its Equivalent Classes to the Client.

- ***Find a service***

This method is used if the Client wants to find a service but knowing very little information about the service (for e.g. just having a Keyword) as illustrated in Figure 41.

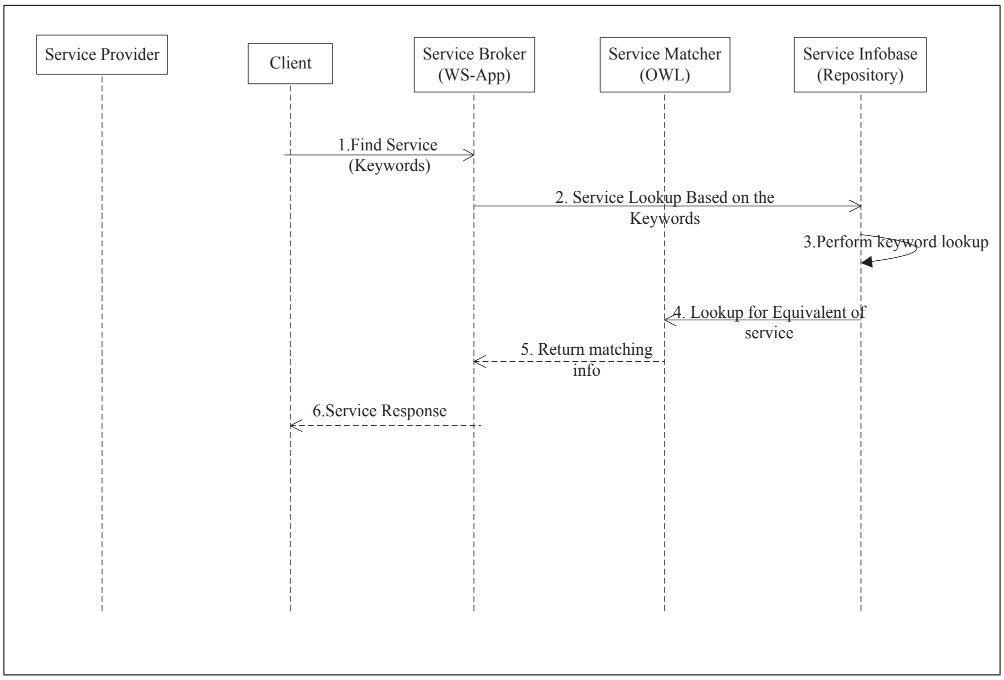

Figure 41 Call sequence diagram for Find a service

i) The Client can also find a service via the Keyword if he is not sure about the service Name.

ii) The Service Broker will do the Service lookup based on the Keyword and send it to the Service Infobase.

iii) The Service Infobase will then perform the Keyword lookup.

iv) The Service Matcher (OWL) will then lookup for Equivalent services.

v) Service Matcher will return the matching information to the Service Broker.

vi) The Service Broker will finally send the service response to the Client.

D. System user interface and use scenarios

In this section, the future service discovery system user interface is explained and based on the requirements specified in Section 3.3.1, three use scenarios are selected for description in detail.

- ***System user interface***

Figure 42 Future service discovery system user interface

Figure 42 depicts the user interface for future service discovery systems. It has three main methods which are:

i) *Register Service* – for the user to register a new service. The fields such as Service Name, Service URI, Service Description, Keyword, Service Type and Available Services are required to be filled.

ii) *Register Service Type* – during registration of a new service, if the user finds that the service type of the registered service is not yet available, he may needs to register a new service.

iii) *Find Service* – there are three options to find service for the user. For finding a service by service name, the EquivalenceClass semantic will be used and all the similar services with different names and languages will be returned. If the user chooses to find service by the service type, all the subtype services under the same ParentType will be returned and if the Keyword is used to find service, even the non-exact match services will be returned as long as it has something in common to describe about the service (for e.g. Snacks or Cocktail to define Food service).

- ***Use scenarios***

Three use scenarios are presented to illustrate the applicability in practice which are:

i) Use scenario 1 - The network system has similar services with different names and different languages.

When a user wants to find a service, such as - a Restaurant service he will use either standard English or his native language. The Restaurant service can be registered under other different names in multiple languages for e.g.

Restaurant: {Restoran, Ristorante, Café, Bistro, Warung, … }

If the user enters the word Café, the future service discovery will return not only the Café service but also other similar service with Café in other names as depicted in Figure 43.

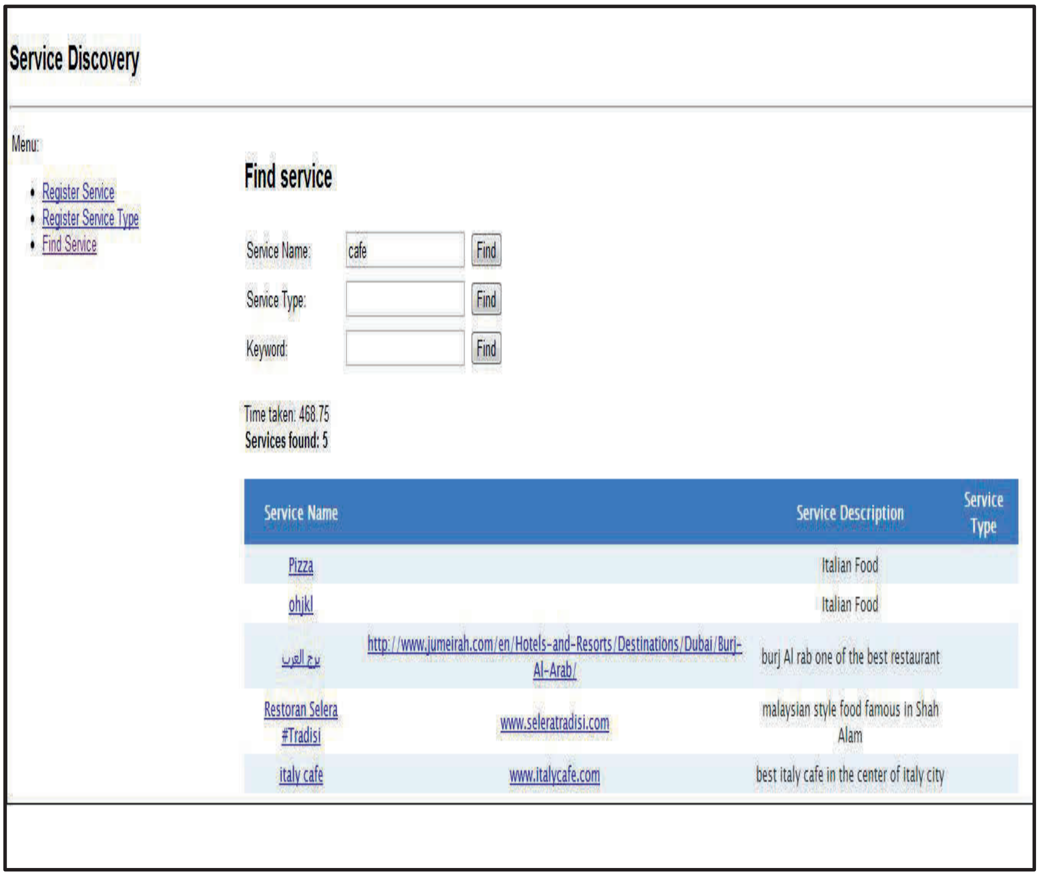

Figure 43 Use scenario 1a) – having similar service with different names

In addition, the future service discovery also supports finding a service in multiple-languages as shown in Figure 44.

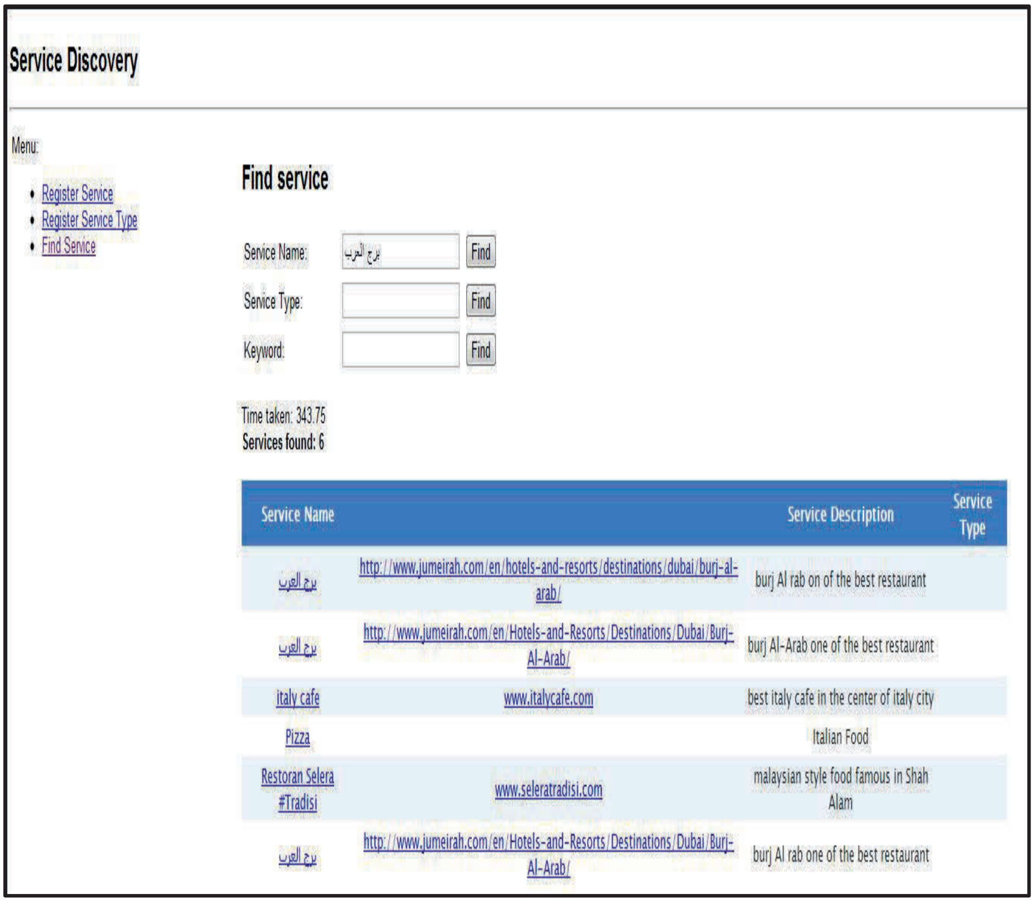

Figure 44 Use scenario 1b) – having similar service with multiple languages

By having this feature, the user will not have to worry even though other names returned when he wants to find a service (especially if he is abroad and different languages are used to call a service) because the service discovery system returns only services equivalent to the one requested.

There are three options for the user when it wants to find a service:

- by Service Name – if the user knows exactly the service it wants to look for

- by Service Type – if the user is not sure about the Service Name but only knows the Service Type
- by Keyword – if the user only knows very little about the service and has some clues describing the service (for e.g. Dining or Meal which refers to Restaurant)

ii) Use scenario 2 - The network system has different services with the same name.

Since in the future ubiquitous communication systems the service can be anything and introduced by anybody, there can be a possibility of having different terms and perspectives of knowledge of the service from the user and service provider. The future service discovery supports the function of having different services with the same name as depicted in Figure 45.

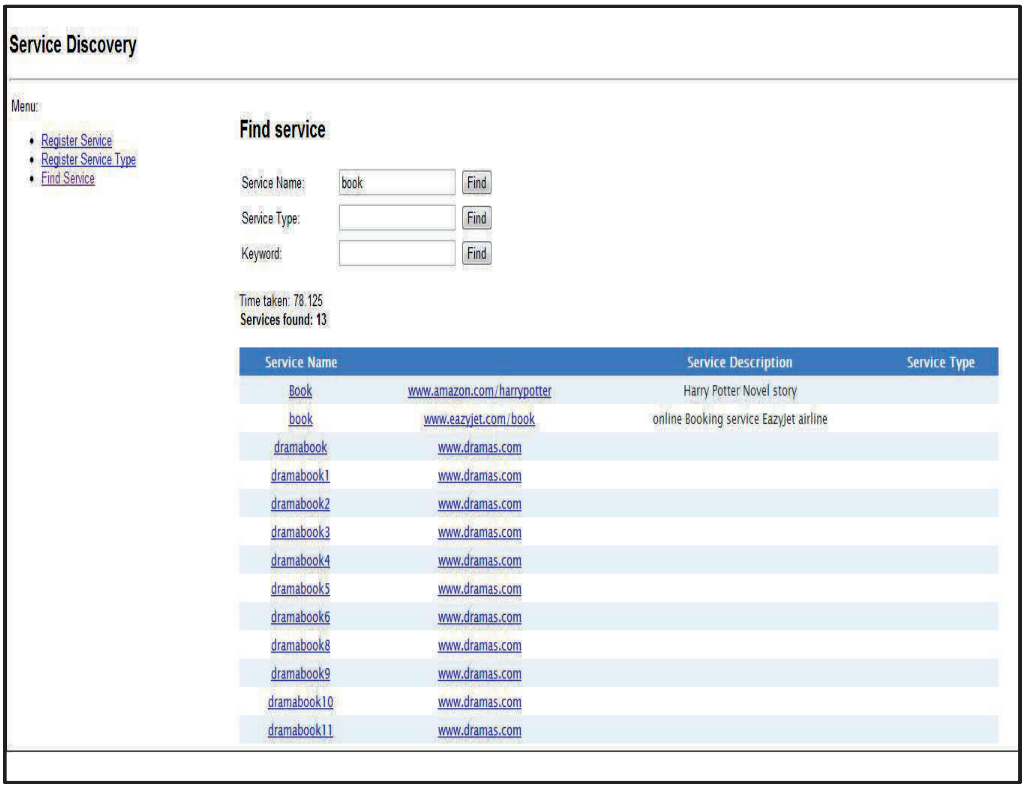

Figure 45 Use scenario 2 – having different services with the same name

As in the above example, the word Book can refer to a type of Book service (for e.g. buying online books or information about a book). There can also be other meanings of Book service which may refer to a Reservation service.

Even though the future service discovery allows having different services with the same name, the ambiguity and confusion can still be avoided via the details of the Service Description and Service Type returned during the service search.

iii) Use scenario 3 - The service discovery returns partial matches to requested service.

The future service discovery is using a semantic matching in addition to syntactical matching as the one used in the existing service discovery. This is very important especially in the situation where no equivalent service available but there exists service, which have additional functions to the requested ones. Such a service is called – partial match.

The future service discovery will introduce the use of service subtyping by having ParentType and attributes ParentType in the Service Type description template. As illustrated in Figure 46, when the user is asking for Telephony service; Skype, SIP and G1 are returned because they are grand children of Telephony (Skype and SIP are subtype of IP Telephony while G1 is subtype of GSM) and have inherited all the characteristics of Telephony. In this case the Skype, SIP and G1 are having similar functions (voice call – which is the generic Telephony features) but some of them have more or different additional functions (for e.g. Skype has a video call feature) and they are also different in service components and service implementations.

Figure 46 Use scenario 3a) – having partially match service

However, if the user is asking for a specific Skype service, only the Skype is returned and nothing else. This is as illustrated in Figure 47.

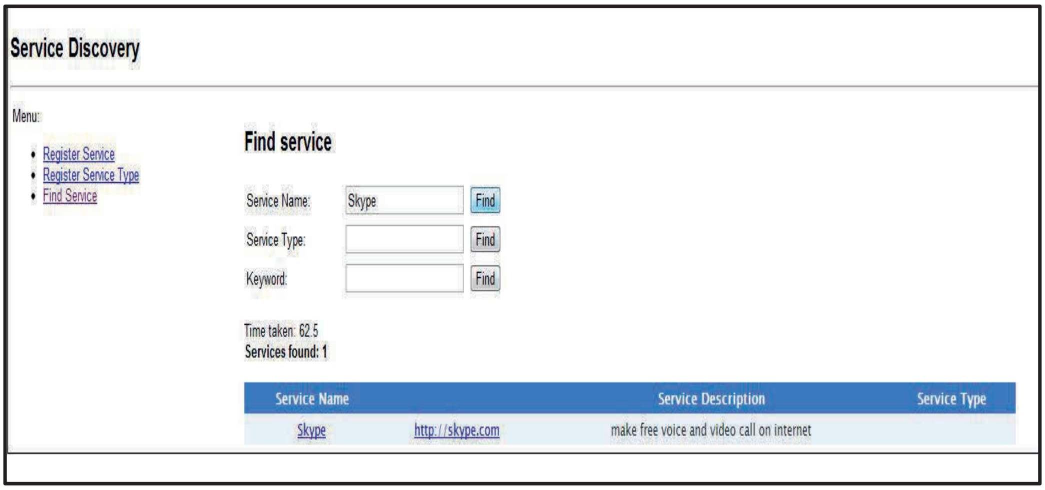

Figure 47 Use scenario 3b) – service subtype

E. System performance

As in the future mobile environments the performance of service discovery in dynamic changes is a major concern of the user. It is very important to ensure that the service requested is returned in an acceptable time (fast service discovery). Hence performance evaluation is conducted and Figure 48 shows the result of the testing which is done based on the four different search criteria:

i) find all available service
ii) find by service name
iii) find by service type
iv) find by keyword

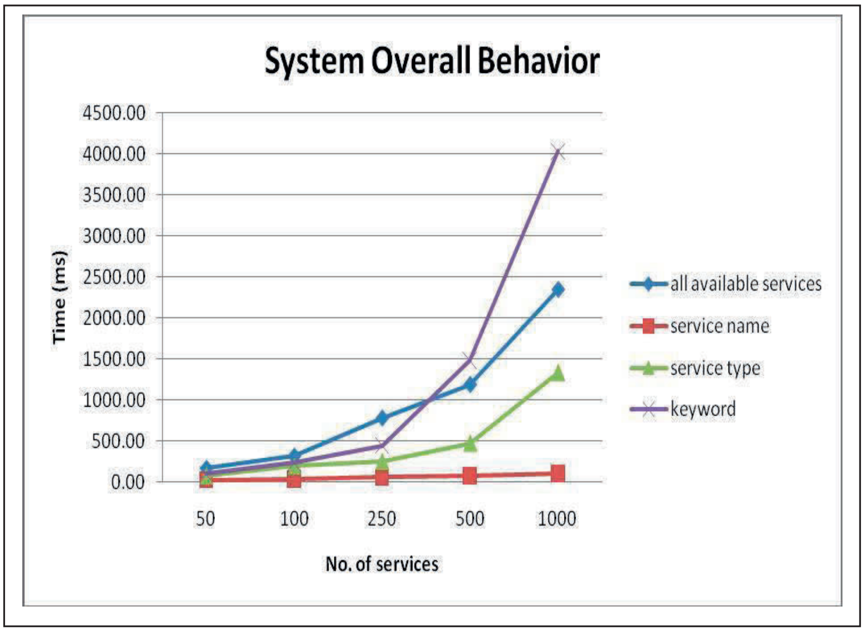

Figure 48 Future service discovery system performance

The x-axis shows the number of services used for testing while the y-axis is the time taken (in milliseconds) to return the results based on different types of parameters used in discovery of services. To measure the time the user was asked to start an Android timer upon issuing a request and stop it when receiving the response.

The mean reading value for each number of service tested was used to plot the graph hence this result is retrieved. Service name lookup was really fast and scales well to real time tests that were conducted at different intervals by adding more services in the system repository. However the service lookup and key word look were considerably slower then preceding service name lookup as after retrieving the information from the database, lookup was carried out in ontologies to provide semantic meaning to the search and service discovery, hence making it gradually slower. Thus the system overall real time performance was found to be satisfactory and acceptable and it scales well to the increased number of services.

3.3.5. Implementation of Service Discovery

In this section, the service discovery development environment is described. The user interface is explained and four use scenarios are given to illustrate the effectiveness of the proposed service discovery and matching procedure.

A. Mobile Service Discovery Testbed

The service discovery application is developed in the Java^{TM} [60] language using the Android Software Development Kit (SDK) which provides the tools and APIs necessary to begin developing applications on the Android platform. Android [47] is a software stack for mobile devices that includes an operating system, middleware and key applications. Android platform is chosen because it provides access to a wide range of useful libraries and tools that can be used to build rich applications.

Eclipse Helios [61] is used for the IDE and Android plugin for Eclipse is also needed for the system development. The developed application is flexible and capable of handling all the request variants from the Mobile Node. Android emulator is used for the testing purpose. The application can be installed and run in any type of mobile phone with Android 2.3 operating system and later.

B. System User Interface

Figure 49 shows the icon of the service discovery in the Android emulator. Figure 50 depicts the service discovery user interface offering three options to find service as follows:

1. by Service Name - the EquivalenceClass semantic will be used and all the similar services with different names and languages will be returned.
2. by Service Type - all the subtype services under the same ParentType will be returned.
3. by Keyword - even the non-exact match services will be returned as long as it has something in common to describe about the service (for e.g. Snacks or Cocktail to define Food service).

Figure 49 Service discovery icon on Android emulator

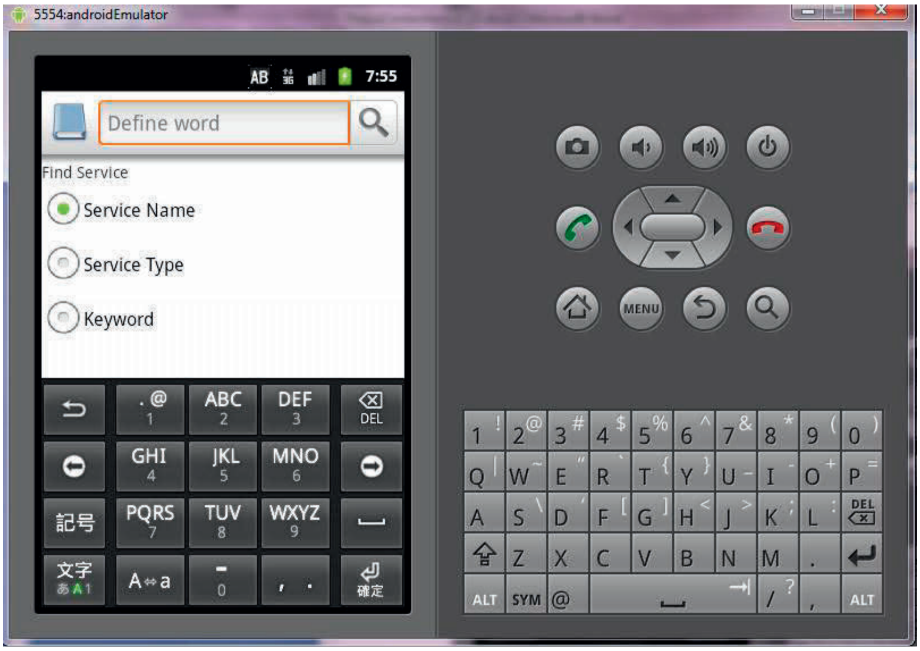

Figure 50 Service discovery user interface

C. Use scenarios

Use scenario 1 - Service discovery and matching based on the service name.

The Mobile Node may request for a specific service by specifying its service name. For example, if the Mobile Node requests for a Skype service, only the networks offering Skype service should be returned. It is worth noting that a network with a completely different service with the name of Skype can be returned because the service names are not standardized and unique. To avoid this situation the Mobile Node may have to specify the service type in addition to the service name to aid the service discovery. For example, when the Mobile Node requests a service with the service name Book and specify Online Business as service type as illustrated in Figure 51 only the Book service with the service type Online Business is returned.

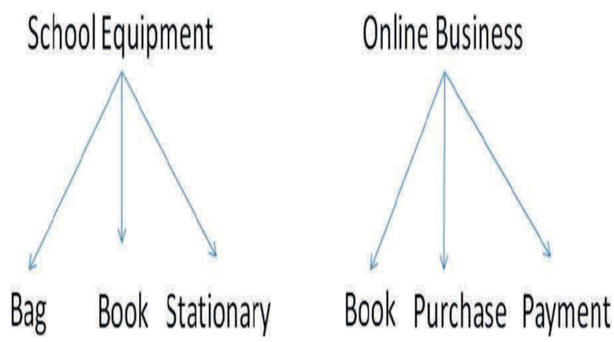

Figure 51 Example of Book service denotes different type of services.

If the Mobile Node specifies only the service type without the service name all the supported services with the same service type will be returned. For example, if the Mobile Node requests for IP telephony service, Skype, VoIP and WiPhone, which has IP Telephony as same service type will be returned.

Use scenario 2 - Service discovery and matching based on ParentType.

The Mobile Node may also request a service by specifying the ParentType. All the services with the same ParentType will be returned. In this case only partial match is required. For example, if the Mobile Node specifies Telephony as ParentType all the child service types of Telephony e.g. Skype, GSM, VoIP and WiPhone will be returned as illustrated in Figure 52.

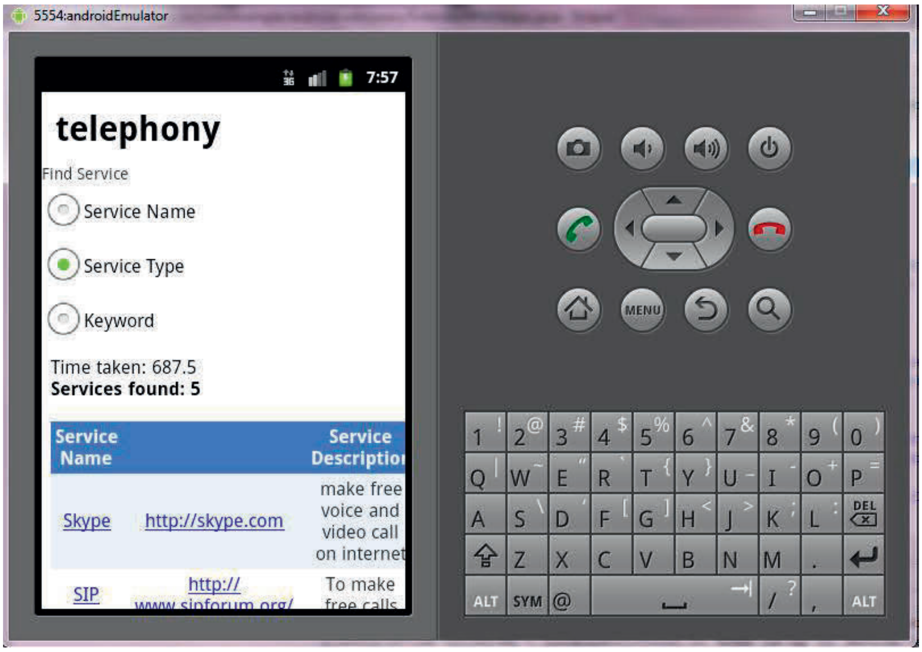

Figure 52 Service discovery and matching based on ParentType

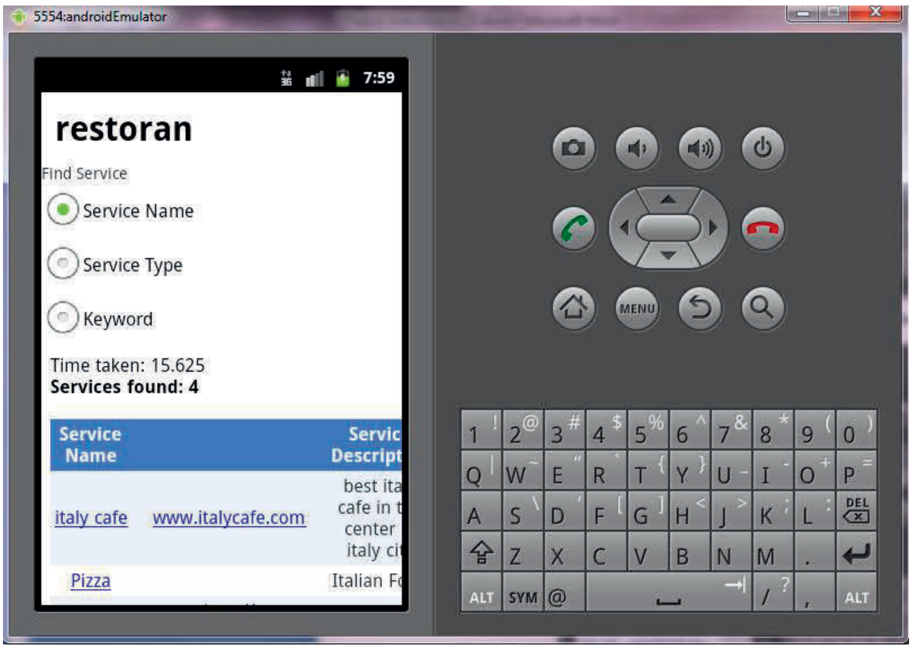

Figure 53 Service discovery and matching based on EquivalenceClass

Use scenario 3 - Service discovery and matching based on EquivalenceClass.

The Mobile Node may request a service by specifying the EquivalenceClass. In this case all the services of equivalent service type will be returned. The returned services can have different names, types in different languages. For example, if the Mobile Node requests service by specifying Restoran as EquivalenceClass, Café, Bistro, Restaurant, etc. will be returned as illustrated in Figure 53.

Use scenario 4 - Service discovery and matching based on Keywords.

The Mobile Node can request a service by specifying Keywords. This can be used especially when the user has only a vague idea about the service he wants to request. The service matching the Keywords will be returned. For example, if the Mobile Node wants to find a Food service, it may send Meal, Snacks or Dining as Keywords to request for the Food service as illustrated in Figure 54.

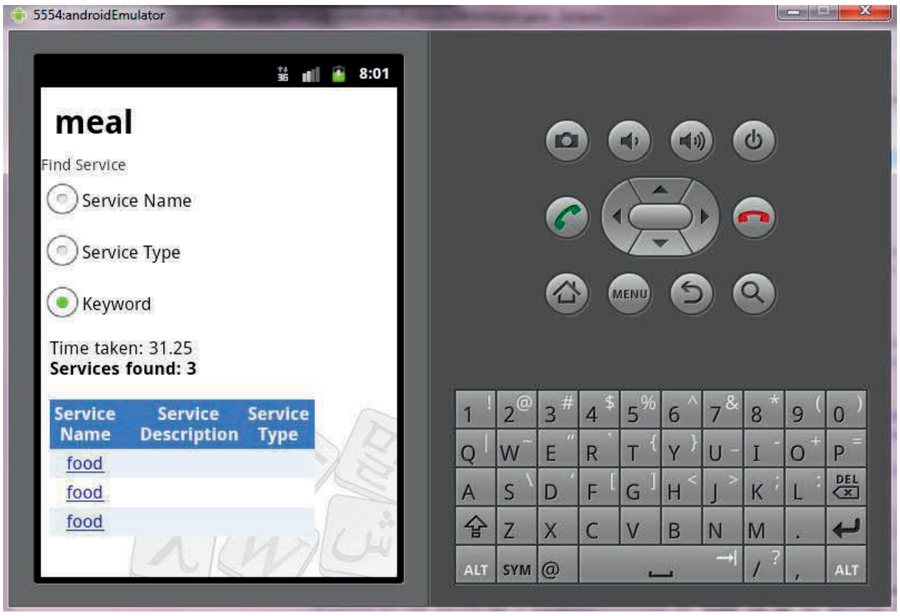

Figure 54 Service discovery and matching based on Keywords

Description of the work

The time taken in the output screen indicates the processing time (in seconds) to retrieve the search result.

4. Conclusion

4.1 Achievements and results

Lately, mobile communications have evolved from homogeneous networks, single access technology and single operator to heterogeneous networks, multiple access technologies and multiple operators environments. This paves the way for more possibilities as well as for more challenges. The first major challenge is to provide the users with a wider and more diversified set of services offered by these multiple heterogeneous networks, owned and managed by different operators. The second challenge is to offer a greater user experience, i.e. user interface, better quality of service, seamless service continuity, etc. The existing mobile service architecture do not take into account heterogeneity and multiplicity and are hence not capable of fulfilling the needs and expectations of the future mobile environments.

To remedy the situation this thesis has analysed future mobile scenarios and propose a mobile service architecture, which enhances the availability of services at the same time as service continuity is ensured. The necessary functional entities both on the network and terminal side are identified and described. The thesis has also conducted thorough analysis and realisation of the two main elements of the mobile service architecture, namely Service continuity and Service discovery.

To realise Service continuity, it is proposed to extend the functionality of the IEEE Media Independent Handover (MIH). More specifically, it is proposed to extend the information database with service information and to include service information in the information exchange between the mobile client and the MIH.

Regarding service discovery, an innovative discovery scheme capable of handling the following characteristic is proposed:

- Similar services with different names in different languages
- Different services with similar names
- Partially similar services
- No restriction regarding the name length
- Anybody can introduce anything as a service anytime

To verify the validation of the three proposed items:

- The mobile service architecture
- The Service Continuity
- The Service Discovery

partial but sufficient implementations have been carried out. Consequently, it is possible to conclude that the three proposed items are both sound and feasible.

4.2 Critical review of the work

In this section a critical review of the three contributions of the thesis, namely the mobile service architecture, the service continuity and the service discovery is provided.

4.2.1. The mobile service architecture

Although the mobile service architecture proposed in this thesis is a pure technical solution, which puts the user in the centre, it allows the mobile user to discover new and consume services as soon as they appear. The solution does not consider the political and business aspects which involve other parties such as network operators and service providers. In fact, it is not evident that a service provider is willing to let a user change to another service provider. In order to be accepted, it might be necessary to elaborate a sound business model which identifies the incentives and benefits of all the parties. A balanced charging scheme and a fair revenue sharing model are also required. Last but not least, there is a need for a new identity management system, which provides adequate authentication and protection of the user while preserving flexibility and allowing single-sign-on.

4.2.2. The service continuity

To realise service continuity extensions to the IEEE 802.21 MIH has been proposed in this thesis work. The proposal has been published as papers at international conferences but it is not submitted to the standardisation body in charge, namely IEEE. Due to the lack of resources and time, it is not possible for the author to participate to the IEEE and submit the proposal. It is always a hope that the idea is adopted by other persons or organisations that will again promote it at IEEE to be approved as standard. Furthermore, components on the mobile device side must also be standardised and should be submitted to standardisation bodies like Open Mobile Alliance (OMA), W3C, etc.

4.2.3. The service discovery

The feasibility of the proposed service discovery has been verified by an implementation using Web service and mobile phone technology and based on EquivalenceClass, ParentType and Keywords. The implementation is, however, very simple and therefore does not prove the usability of the service discovery. It is still unsure that the service discovery can be accomplished within an acceptable amount of time when the number of available services is very large or when the set of equivalent services is getting big. Furthermore, it is not verified that the service discovery always return correct results, i.e. return the services that the user is looking for. It is neither verified that the service discovery is capable of finding all the services equivalent with the one specified by the user when the number of equivalent is large.

4.3 Further work

Several tasks could not be accomplished during the thesis work due to the lack of time and will be described in the following sections.

4.3.1. Service continuity

As further work, the service information and the extension of handover procedure in the existing IEEE 802.21 MIH Services proposed in this thesis work should be implemented and integrated in an 802.21 MIH implementation. An extended MIH client should also be implemented and installed on a mobile platform such as Android, Symbian, Windows 7, etc. Thereafter, testing and evaluation should be carried in the real mobile environment consisting of a mobile device equipped with an extended MIH client and a real extended MIH implementation. Tests should be carried out on various situations such as large number of mobile devices, highly mobile devices, mobile devices with different moving pattern, etc.

It might also be interesting to introduce other criteria for selection and change of service than signal quality such as security, price, bandwidth, etc. The shift from higher performance service to lower one and vice versa is a very exciting and important issue that deserves more investigation.

4.3.2. Service discovery

As mentioned the feasibility of the service discovery is demonstrated but its usability is not yet proven. As further work, it is hence relevant to develop a prototype which incorporates more semantic matching with more level of parents and larger equivalent set. Thereafter, tests should be carried out on situations with large number of services, and large sets of equivalent service to demonstrate that the service discovery can be completed in time. Last but least, it might be relevant to emulate a distributed environment, where a service provider adopts, modifies, registers and deploys services in unorganized and unsynchronized manner. It will be quite interesting to see whether the service discovery manages to find all the services correctly.

The thesis publications

Part II – The thesis publications

Index of Part II

The thesis publications

PAPER 1: Enabling Service Continuity on Future Mobile Services

Nor Shahniza Kamal Bashah, Ivar Jørstad & Do van Thanh

Proceedings of 4th International Symposium on Wireless Pervasive Computing (ISWPC2009), Melbourne, Australia, February 11-13 2009

ISBN 978-1-4244-2966-0/09

Enabling Service Continuity on Future Mobile Services

Nor Shahniza Kamal Bashah*
Department of Telematics, Norwegian University of Science and Technology, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway
nor@item.ntnu.no

*Ivar Jorstad***
Ubisafe AS
Gamleveien 252, NO-2624 Lillehammer, Norway
ivar@ubisafe.no

Do van Thanh***
Department of Telematics, Norwegian University of Science and Technology - Telenor R&D, Snaroyveien 30, NO-1131 Fornebu, Norway
thanh-van.do@telenor.com

Abstract

To be able to access services in a seamless way across different physical access networks is compelling. The seamless handover features and availability of the mobile voice telephony service is by far the most important key to success in current mobile telecommunication services. As mobile devices are becoming multi-modal, i.e., supporting several physical network access technologies, there has been much focus on enabling network handover between these access technologies. Unfortunately, seamless handover is not sufficient to ensure service continuity because in many cases, due to mobility not only the connection link must be changed but also the components realizing a service. In this paper extensions to the 802.21 MIHF to provide adequate assistance to the dynamic composition of a service is proposed and described thoroughly.

1. Introduction

Seamless handover in horizontal network (i.e., change from one base station to another one base station in GSM network telephony) is a well-known technology. Seamless handover in heterogeneous network (i.e., changing from GSM network to WLAN) is a more challenging issue since it requires continuous scanning of a broad spectrum of frequencies using different coding schemes, which is very power consuming. With the introduction of IEEE 802.21 Media Independent Handover Function (MIHF), the mobile device can perform handover while consuming minimum energy. Its interface, Service Access Points (SAPs) enables the mobile device to have continuous access to the network. Information about neighboring networks is shared through the MIHF, a shim layer in the mobility-management protocol stack of both the mobile node and the network elements. It helps mobile device to discover, characterize and select network within their current neighborhoods by exchanging information and defining commands and events to assist in handover decision making process. Unfortunately, seamless handover is not sufficient to ensure service continuity because in many cases, due to mobility not only the connection link must be changed but also the components realizing a service. To give adequate assistance to the dynamic composition of a service there is a need for extensions in the MIHF.

This paper starts with a brief introduction about IEEE 802.21 Media Independent Handover Services in Section 2, followed by the highlighted of the limitation of the current Media Independent Handover Function (MIHF) in Section 3. Section 4 identifies the requirements from future mobile services and Section 5 continues with the proposed extensions in MIH. Finally Section 6 concludes the discussion.

2. Short introduction about IEEE 802.21 Media Independent Handover Services

From a terminal connected to a dedicated wireless network mobile devices have evolved to be multi-system and multi-standard terminals equipped with several wireless access technologies such as Global System for Mobile Communications (GSM), 3rd Generation Wireless Communication

(3G), Wireless Local Area Network (WLAN), Bluetooth, etc. The connectivity is supposed to be considerably improved but without coordination, synchronization and handover mechanisms, it is difficult to make use of all these available wireless technologies to provide continuous connectivity. In fact until now handover is only possible between Base Stations or Access Points of same technologies.

To remedy this situation, IEEE has introduced a new standard called 802.21 that is aiming at facilitating handover between heterogeneous networks [1] for devices equipped with multiple wireless access technologies.

The handover process consists of three phases:-

- Handover initiation which is searching for a new link. It is comprised of the following tasks:
  - Network discovery
  - Network selection
  - Handover negotiation
- Handover preparation which is setting up a new link. It is comprised of the following tasks:
  - Layer 2 connectivity
  - IP connectivity (layer 3)
- Handover execution which is transferring the connection. It is comprised of the following tasks:
  - Handover signaling
  - Context transfer
  - Packet reception

The IEEE 802.21 assists the first two phases, handover initiation and handover preparation and leaves the handover execution to existing protocols. In fact, the most challenging task in vertical handover, i.e. across heterogeneous technologies over different access networks [2] is the network discovery which requires continuous scanning of a broad spectrum of frequencies using different coding schemes. This is a very energy consuming task that a mobile device operating on battery cannot afford. The IEEE 802.21 provides Service Access Point (SAPs) on each access network which can be queried by upper layers (layer 3 and higher) to get information about the network availability and Quality of Services (QoS) at each time. The mobile device can hence perform handover while consuming minimum energy.

The IEEE 802.21 supports the handover process for both mobile and stationary users. For the mobile users, the handover process could happen due to changes in wireless link condition or due to the gaps in radio coverage while the mobile terminal is moving. As for the stationary users, the handover process might occur due to the changes of the surrounding environment such as having another network which is more attractive than the other network or the needs to change to another network due to the network requirement such as a need to have a higher data rate channel during downloading a large data file.

The IEEE 802.21 introduces a logical entity called Media Independent Handover Function (MIHF) which is a shim layer in the mobility-management protocol stack of both the mobile node and the network elements. It facilitates the handover decision making based on the upper layer decisions and link selection based on inputs and context from Media Independent Handover (MIH).

Figure 1 shows the components of MIH and how its function.

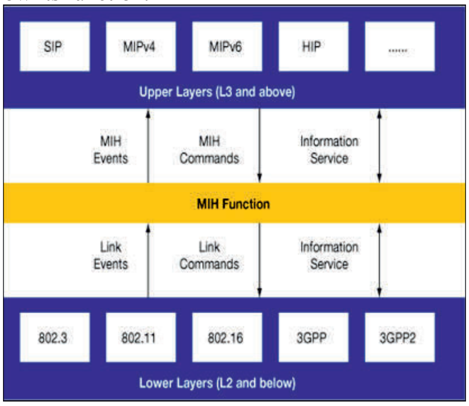

Figure 1. IEEE 802.21 MIH architecture

Basically the MIHF consists of three major services which are Media Independent Event Service (MIES), Media Independent Command Service (MICS) and Media Independent Information Service (MIIS).

MIES functions are to detect events and deliver triggers from local and remote interfaces to the upper layers. It handles all types of event classification, event filtering and event reporting from the dynamically changes of the link characteristics, link status and link quality.

The MICS provides a set of commands for MIHF users to control handover relevant link states. The commands are sending to the lower layers to determine the status of the connected links and also to execute mobile and connectivity decisions of the higher layers to the lower layers. This service controls and manages link behavior relevant to handovers and mobility.

MIIS provides details on the characteristics and services provided by the securing and surrounding network. It provides a framework by which a MIHF in the mobile node and in the network can discover and obtain homogeneous and heterogeneous network information within a geographical area to facilitate handovers. With that, the network elements will receive the information about neighboring networks, thus enabling making of more effective handover decisions across heterogeneous networks.

To illustrate the role of IEEE 802.21 in handover let us consider the case when the mobile phone is moving from a GSM network to WLAN.

Currently without the MIH function, there is no way that the mobile device can detect WLAN coverage because the WLAN interface is usually disabled due to battery drain. To use WLAN, the user has to switch the interface on. No service continuity is supported.

With the IEEE 802.21 the mobile device will have continuous access to the MIH function via a SAP on the GSM network. When entering a WLAN area the MIIS will detect it. The MIES will be used to send an event MIH_Link_Detected to the upper layers on the mobile terminal e.g. Mobility manager, which can turn on the WLAN interface.

The Mobility manager can also issue the command MIH_MN_HO_Candidate_Query to obtain handover related information about possible candidate networks.

It can then make the MIH_MN_HO_Commit to notify the network that a candidate. Thereafter, it can initiate the handover by performing the necessary signaling. When handover is completed, the mobility manager will issue a MIH_MN_HO_Complete to indicate the handover has been completed.

The handover process is as illustrated in Figure 2.

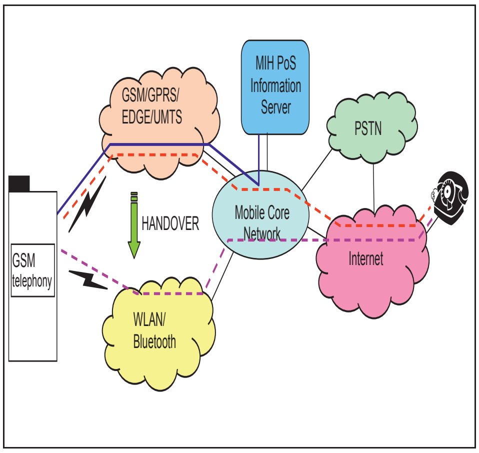

Figure 2. Handover process with IEEE 802.21 Information Service

3. Limitation of the 802.21 Media Independent Handover Function (MIHF)

As described in the previous section, the 802.21 Media Independent Handover Function (MIHF) does successfully support handover between heterogeneous access networks that are incorporated in the same core network as specified by 3GPP in [3]. The access networks may belong to the same network operators or different ones. Service continuity is obtained with a link through the new access network.

It is worth noting that the necessary condition for success is that the same service implementation is used across all heterogeneous access networks. In the previous example, GSM telephony service is used both in the mobile access networks and in the WLAN network.

It is more challenging to realise service continuity when the mobile phone is moving from a GSM coverage area into a WLAN area which belongs to an IP network operator and does not have any connection to the 3G network. The GSM telephony service is not available in such area but there is an IP telephony service. Even with the assistance of the MIHF, i.e. detection of the WLAN access network, it is not possible to carry out handover because there is no way the mobile phone can perform the necessary signalling to establish a new link to transport the GSM voice traffic. Service continuity cannot be ensured in this way.

The implementation of service continuity for the telephony service in this case requires not only a link change but also the change of the telephony service from GSM telephony service to IP telephony service. To enable a fast "service handover" the following tasks must be accomplished:

- Service discovery
- Service matching
- Initiation of the service client of the mobile terminal
- Registration of the service client to the service server
- Signalling and establishment of new links
- Service switching

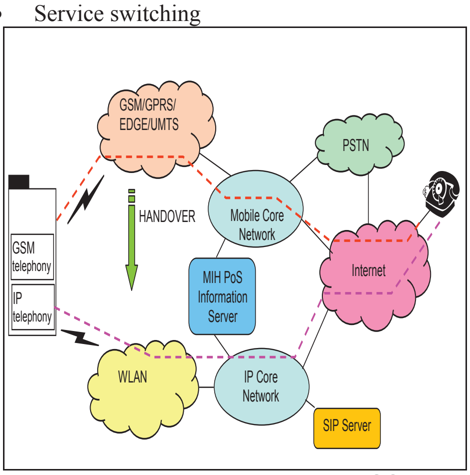

Figure 3. Handover between GSM telephony and IP telephony

The service discovery can, of course be carried out by the mobile terminal but will be more efficient with the assistance of the 802.21 MIHF; something which is not done today.

To elucidate the limitation of the 802.21 MIHF let us consider again the situation of the mobile phone moving from the mobile access network to a WLAN area belonging to an IP network operator.

When the mobile phone moves in the WLAN coverage area, the MIIS will detect it. The MIES will be used to send an event MIH_Link_Detected to the upper layers on the mobile terminal. The service continuity server can ask for a list of services available in the area which includes the IP telephony service. The service continuity server can perform a service matching and find that the IP telephony service is of type telephony and is equivalent to the GSM telephony service. It is hence possible to do a service handover.

The service continuity server can then initialize the IP telephony client on the mobile terminal. The IP telephony will then proceed with the registration to the sip server and carry out the necessary signalling to establish a new voice link. To complete the service handover the GSM voice link will be terminated.

4. Requirements from future mobile services

The limitation of the IEEE 802.21 is not confined to the case described in the previous section but applies to generic future mobile services. Indeed, a future mobile service will be realized by several equivalent implementation alternatives using different technologies and having different service components. When the mobile device is moving it might be necessary to change the implementation alternative and its components due to different reasons such as air link availability, service implementation availability, quality of service, security, pricing, etc. An example of service change is the case of telephony service described in earlier section which is realized first by GSM telephony and then IP telephony. To ensure service continuity, it is necessary that a change of service implementations is carried out in addition to a change of connected links.

More generally, the following statements can be formulated:

- A mobile service can be realized by one or more service implementations
- A service implementation consists of one or more service components
- Service continuity is ensured by both a change of connection links and a replacement of one service implementations with another one

From the statements above, three cases can be derived:

Case 1: Change of links only (see Figure 4)

Case 2: Change of links and service logic components in the network (see Figure 5)

Case 3: Change of links and service logic components both in the client and network (see Figure 6)

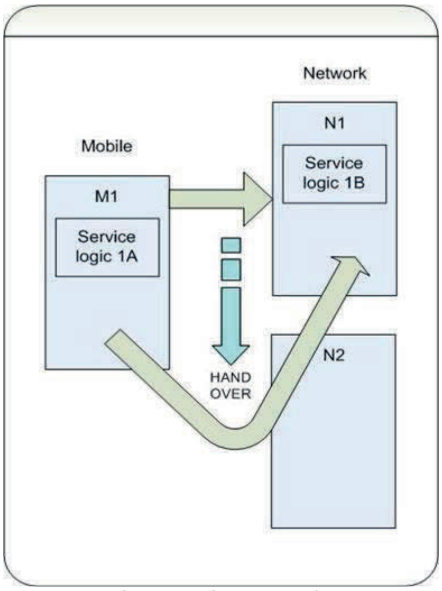

Figure 4. Case 1: Change of links only

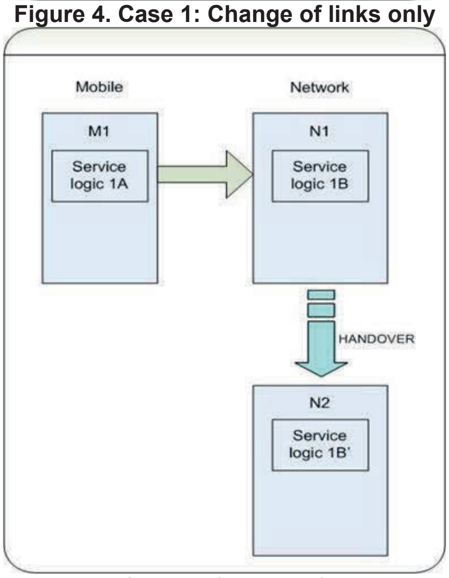

Figure 5. Case 2: Change of links and service logic components in the network

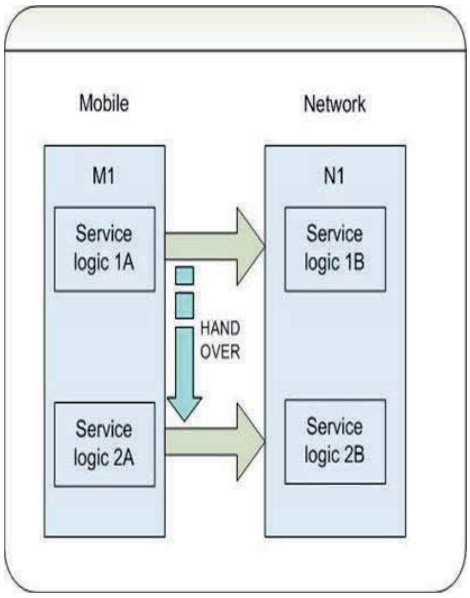

Figure 6. Case 3: Change of links and service logic components both in the client and network

As depicted from the figures above, the user can still enjoy the service (in this example, a telephony service) provided that it is realized by the equivalent service logic components in both the mobile and the network side.

In order to support service continuity, the future mobile service environment must support the dynamic service composition, i.e. allowing a service to be composed dynamically by equivalent service logic components. Indeed, there is a need to ensure that the realization of the service can be done by dynamically discovering, selecting, assembling and executing the service logic components. In order to realize the service, there is a need for information about service components such that identification and matching of service components can be done.

To determine the semantic equivalence of two services and the semantic equivalence of the service components, it is needed to have a precise definition of services, service components and service ontology. As explained in [4], the semantic equivalence of two services S_1 and S_2 is determined by:

1. Semantic equivalence of the service concept
2. Semantic equivalence of the interfaces

And in this example S_1 ***SE*** S_2 which states that S_1 is semantically equivalent to S_2.

Finally to use the service components, the syntactic properties of the services should also be determined.

5. Proposed extensions in MIH

In order to ensure service continuity, it is necessary that the mobile device has the following information:

1. What services are available at every time
2. Which ones are the ones that it wants to use
3. How to use these services

Briefly, the mobile device must have access to sufficient information to perform:

1. Service discovery
2. Service matching
3. Service initiation

The current 802.21 MIHF provides the upper layers on the mobile node with information necessary to perform handover, i.e. the change of the connection links but does not supply any information about services. Therefore, in this paper, we propose to extend the information service of the MIHF to cover service information which is required in the service discovery, service matching and service initiation.

5.1. Service Discovery

Service discovery is a process of obtaining a set of services which can possibly fulfill a user request [5]. Thus, the devices may automatically discover network services including their properties, and services may advertise their existence in a dynamic way [6]. The primarily service discovery is limited only to a specific geographical scope and in a fixed infrastructure. The discovery is just in syntactic level and use the attribute-value scheme hence sometimes may give an irrelevant match to the result. Some of the service discovery protocols introduced are such as Jini [7], UPnP [8], Salutation [9] and Service Location Protocol (SLP) [10]. However, as the mobile services growth in a dynamic environment, there is a need to have a service discovery in a semantic level. The semantic service discovery is using an ontology-based approach which can provide an ontological framework that is capable to describe and process the services. In order to implement the service discovery there are few items needed which are:

5.1.1. Service Naming. Service naming is a method on how the service is called. To support service invocation and access to resources, a service or resource must be uniquely identifiable and addressable [11]. In order to do that, a name is assign to the service and name resolution will map the name to an address. Through the service naming, it will allow for abstracting the identity of a service or resource from its location and access mechanism.

5.1.2. Service Classification. Service classification is a process of assigning a class to a service, indicating the domain/category a service belongs to.

5.1.3. Service Listing. The classification of the services will then be detailed out by the service listing. An example of service classification and service listing is as illustrated in Figure 7.

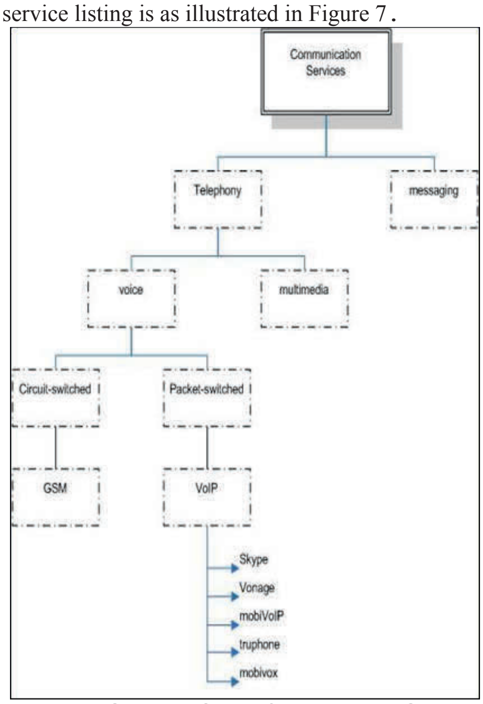

Figure 7. Service Classification and Service Listing

Communication services is the unique name that identifies the service. It is then classified into two categories which are telephony and messaging. The telephony is classified into two which are voice and multimedia. Under voice classification, there are circuit-switched and packet-switched. GSM is an example of voice telephony using circuit-switched while Voice over Internet Protocol (VoIP) is an example of voice telephony using packet-switched.

Under the VoIP classification, there are types of services listed which are Skype, Vonage, MobiVoIP, Truphone and Mobivox. During the service discovery, all the services under VoIP classification will be listed and user may choose which one that match with their requirement. However, in having the listing of the services there are some factors that need to be considered such as the size of the listing as well as the quantity of the listing. If the listing consists of so many services it will increase the size and the quantity of the classification and may result inefficiency in service discovery process.

5.2. Service Matching

Service matching is the process of comparing the service request against the available service advertisements and determining which service best satisfies the request [12]. As discussed previously, there are two types of service matching approach which are syntactic and semantic. The syntactic approach service matching will only do the comparison based on attributes or interfaces while the semantic approach requires a service description complete with the ontology definition and a service engine with reasoning mechanism [13].

Referred to [14] the matching process is organized as a series of five filters which are namespace filter, text filter, domain filter, i/o type filter and constraint filter. Each filter is independent of each other. Each of them narrows the set of matching candidates with respect to a given filter criterion.

The matching process should be accurate and fast at the same time. Two factors should be considered which are response time and scalability.

5.3. Service Initiation

As stated earlier, to ensure service continuity the mobile device needs to have the information about the services that are available which is done by service discovery, which services that the mobile device wants to use which is solved by service matching and last one is how to use the service which is through the service initiation. Before the service can be used or initiated there are some information which is necessary such as what are the components of the services are and how to initiate them. As referred to [15], there are seven service composition classes which are activity, condition, event, flow, message, provider and role.

Once the service components have been identified, it is then can be initiated by having the exchange of data or messages between an associations of participants. It is important that they have to agree on a characterization of a session that they would like to share and from there only the service continuity can be ensured.

5.4. Additions to MIHF

Two additional functions have been identified in the current MIHF in order to support service continuity. Figure 8 illustrates the additional functions which are service information and service description.

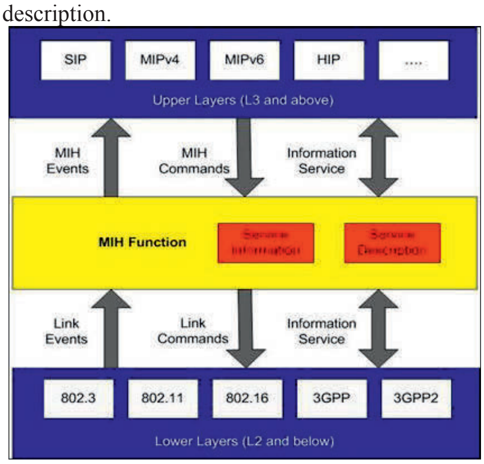

Figure 8. An extension of MIHF

6. Conclusion

The current IEEE 802.21 Media Independent Handover Services facilitate handover across heterogeneous access networks through exchanging of messages by the shim layer, MIHF in both mobile node and network elements. However, it does not support service continuity when the new access network is belongs to different network operators or is not incorporated in the same core network. In this case, service continuity does not require only the change of link but also the change of the service components. In order to ensure service continuity, we identified the need for additional functionality in the current MIHF which are service information and service description. The service information provides details about service components in both the mobile and the network. Details of the service are described in the service description such as service name, service classification and service listing. By using the provided information the service can be discovered, matched and initiated. Further works of this research will be the design and implementation of these functions.

7. References

[1] Media Independent Handover Services, "Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", IEEE P802.21/D7.1, August, 2007

[2] Jared Stein: Survey of IEEE802.21 Media Independent Handover Services - http://userfs.cec.wustl.edu/~jws2/mihs/index.html

[3] Wireless Local Area Network (WLAN) Interworking, "3GPP System to Wireless Local Area Network (WLAN) Interworking; System Description Release 7", Technical Specification, 3GPP TS 23.234 v7.7.0, June, 2008

[4] Jorstad, I. & van Do, T. (2006). "Personalisation and Continuity of Mobile Services", Chapter in Mobile Multimedia: Communication Engineering Perspective", pp. 59-84, Nova Science Publishers, inc. ISBN: 1-60021-207-7, 2006

[5] B. Sapkota, l. Vasiliu, I. Toma, D. Roman & C. Bussler, "Peer-to-Peer Technology Usage in Web Service Discovery and Matchmaking", Web Information System Engineering- WISE 2005, pp. 418-425, Springer Berlin/Heidelberg Publisher, ISSN 0302-9743, Volume 3806/2005

[6] C Bettstetter & C Renner, "A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol", Eunice 2000

[7] Sun, "Technical White Paper: The evolution of Jini technology in Telematics", http://www.sun.com/software/jini, October 2001

[8] Universal Plug and Play forum, "UPnP device Architecture", Version 1.0, http://www.upnp.org, 24 April 2008

[9] Salutation Consortium, "White Paper: Salutation Architecture", http://www.salutation.org

[10] RFC 2608, "Service Location Protocol", Version 2, http://www.ietf.org/rfc/rfc2608.txt

[11] R. Ahmed, R. Boutaba, F. Cuervo, Y. Iraqi, T. Li, N. Limam, J. Xiao & J. Ziembicki, "Service Naming in Large-Scale and Multi-Domain Networks", IEEE Communications Surveys & Tutorials, Third Quarter 2005, Volume 7, No.3

[12] Bandara A, Payne T, Roure D. D & Lewis T, "A Semantic Approach for Service Matching in Pervasive Environments", 2007, http://eprints.ecs.soton.ac.uk/14551/1/ecstr-iam07-006.pdf

[13] Guan T, Zaluska E & Roure D.D, "A Semantic Service Matching Middleware for Mobile Devices Discovering Grid Services", Advances in Grid and Pervasive Computing, pp. 422-433, Springer Berlin/Heidelberg Publisher, ISBN 978-3-540-68081-9, Volume 5036/2008

[14] Kawamura T, Blasio J. A. D, Hasegawa T, Paolucci M & Sycara K, "Preliminary Rreport of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry", Service-Oriented Computing – ICSOC2003, Springer Berlin/Heidelberg Publisher, ISBN 3-540-20681-7

[15] B. Orriëns, J. Yang & M.P. Papazoglou, "Model Driven Service Composition", Service-Oriented Computing - ICSOC 2003, pp 75-90, Springer Berlin/Heidelberg Publisher, ISBN 978-3-540-20681

PAPER 2: Service Naming in Future Mobile Environments

Nor Shahniza Kamal Bashah, Ivar Jørstad & Do van Thanh

Proceedings of 38th International Conference on Parallel Processing (ICPP2009), Vienna, Austria, September 22-25 2009

ISBN 978-0-7695-3803-7

Service Naming in Future Mobile Environments

Nor Shahniza Kamal Bashah
Department of Telematics, Norwegian University of Science and Technology, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway
nor@item.ntnu.no

Ivar Jorstad
Ubisafe AS
Gamleveien 252, NO-2624 lillehammer, norway
ivar@ubisafe.no

Do van Thanh
Department of Telematics, Norwegian University of Science and Technology - Telenor R&D, Snaroyveien 30, NO-1131 Fornebu, Norway
thanh-van.do@telenor.com

Abstract - **Future mobile environments will consist of various network systems that compete to offer services to mobile users. The challenge for the mobile user is to discover the available services and to select the desired services since similar services can have different names and services with same name can be totally or partially different. Furthermore, the service discovery must be accomplished in minimum time to ensure service continuity. A sound service naming scheme is a necessity in future mobile environments. The goal of this paper is to shed light on the service naming area and to propose a service naming scheme which is suitable for a future mobile environment where a mobile device can have connection to several heterogeneous network systems. The paper highlights the definitions used in service naming scheme. Some existing service naming schemes and related works are studied and reviewed. The notion of service is also clarified, which is defined as a work done by the system servers that benefits the end-user. The high-level requirements of the service naming is then identified and based on that eventually a service naming scheme for future mobile services is proposed.**

Keyword - dynamic, service discovery, service matching, service naming, classification

I. INTRODUCTION

Lately, there are emerging mobile phones, called multi-band and multi-mode that are equipped with multiple access technologies enabling connections with multiple network systems. It is hence natural to expect that the connectivity and service availability will be improved. Unfortunately, quite often only one connection with one network system is used at a time e.g. Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), Universal Mobile Telecommunication System (UMTS), Wireless Local Area Network (WLAN), etc. The available services are hence limited to the ones offered by this network system. In a close future multiple connections can be established simultaneously and the mobile device will have access to a wider range of services. The challenge for the mobile device is then to discover the services, figure out their functions, select the desired services, acquire their protocols, syntax and formats and finally invoke the desired services. In other words, the mobile device must know what a service with the name "a" is offering. This could be quite difficult and time consuming when multiple heterogeneous network systems with different concepts and conventions are involved. In fact, the process can be indeterministic and unable to accomplish. It is hence crucial to have a sound service naming scheme. Such a service naming scheme could be incorporated in an extension to the IEEE 802.21 Media Independent Handover Services [1] which is aiming at enabling service continuity as proposed in [2].

This paper starts the review of existing service naming schemes and works. Before starting with the elaboration of the service naming the notion of service is also clarified. Next, a definition of the service naming is given. The high level requirements on the service naming schemes are then described. A service naming will then be deduced and constructed gradually step by step. The paper concludes with some remarks on the proposed service naming.

II. EXISTING SERVICE NAMING SCHEMES AND RELATED WORKS

In order to obtain a sound and usable service naming scheme and to avoid re-inventing the wheel all the relevant existing service naming and related works will now be successively examined thoroughly.

A. Domain Name System (DNS)

Domain Name System (DNS) mapping host names to IP addresses using a predefined hierarchical naming scheme [3]. The highest level is the root of the hierarchy-tree contains number of domains, each of which can contain individual objects (names) and/or lower-level domains. Lower-level domains can in turn have still lower-level domains, allowing the tree as a whole to take on an arbitrary structure.

It is usually consists of two or more segments written separated by dots (e.g., www.ntnu.no) and implements a distributed (keyword-based) redirection service. When the DNS resolver first query, it will interprets the name segment by segment (from right to left) using an iterative search procedure. The rightmost level is the top level domain and each label to the left specifies a subdivision or subdomain of the domain above it. The corresponding DNS server will then provide a pointer to the next server which it should consult. It will then query one of the root servers to find the server authoritative for the next level down. Next, it will query the second server for the address of a DNS server with detailed knowledge of the second level domain and the process continues until the final step. It helps names to be organized from most general to most specific.

The domain name has hierarchic architecture which can be inspired in the service naming for future mobile services since it can contribute to reduce the matching time. Unfortunately, it does not allow any encapsulation of semantic which is necessary in the service matching for service continuity.

B. Service Location Protocol (SLP)

Service Location Protocol (SLP) uses Uniform Resource Locator (URL) to locate the service based on the description of the service requested by the user [4]. It consists of 3 entities which are: User Agent (UA) that initiates service discovery on behalf of clients, Service Agent (SA) that advertises the location of one or more services and Directory Agent (DA) that acts as a centralized repository for service location information.

The Service Location Protocol is focusing only on the discovery, location and configuration of services and does not address sufficiently the service matching. In fact, only the service type does contain about the semantic of the service and it is definitely inadequate in a world where almost everything can be a service which can be defined by anyone.

C. Jini

In Jini, lookup service is used for the services to register themselves before they can be used by the clients [5]. The registration provides information on what are the services they offered and their locations. When a client wishes to use a service, it will send a *service template* containing description of the service that is requested. The lookup service then returns information on matching services.

With the Java RMI as its protocol, Jini supports the interoperability function which is one of the requirements in the service naming for future mobile services. However, it‘s attribute-based search only provides syntactic comparison based on attributes and interface, thus unable to identify equivalent concept.

D. OMG’s Naming Service

The Object Management Group’s (OMG’s) Naming Service maps a name to remote object reference by using the “naming graph” as the directed graph with nodes and labelled edges [6]. A naming graph allows more complex names to reference an object. Given a context in a naming graph, a sequence of names can reference an object. This sequence of names (called a compound name) defines a path in the naming graph to navigate the resolution process.

The architecture of OMG’s naming service fulfils partially the requirements of service naming for future mobile services because it supports distributed and heterogeneous implementation and requires no dependencies on other interfaces which allow interoperability with other existing service naming schemes. However, it’s only use to locate the service based on the naming contexts hence does not sufficient enough to do service matching in future mobile services where anybody can define anything as a service.

III. CLARIFICATION OF THE SERVICE NOTION

Before proceeding with the specification of the service naming it is essential to clarify the notion of service. There is currently no consensus on the definition of a service. In this paper, a service is defined as a work done by the system servers that benefits the end-user.

The system can include entities like user’s devices, the network and all the available servers. A service is hence realized by hardware and software components located at the mentioned entities.

It is envisaged that in the future anything can be a service and it is consequently impossible predict the number or the nature of the services. There is no consensus on how services should be classified.

However, in order to specify a sound service naming scheme it is necessary to have a simple classification of services as follows:

- **Communication services:**
  - Synchronous: telephony, multimedia telephony, conference, etc.
  - Asynchronous: Short Message Service (SMS), Multimedia Messaging Service (MMS), Instant Messaging (IM), e-mail, etc.
- **Information services:**
  - Browser
  - Search
- **Input services:**
  - Read
  - Scan
- **Output services:**
  - Display
  - Printer
- **Commerce services:**
  - Visit net shop, select and pay
- **Banking services:**
  - Access to bank account
- **Travelling services:**
  - Flight
  - Car rental
  - Boat
  - Taxi
  - Bus
  - Cinema
  - Restaurant
  - Hotel

For each service type there may be defined sub-services as well. For example, the restaurant service may have but is not limited to the following sub-services:

- **List:** to list all the restaurant at a location
- **Find:** to find all restaurants or a particular type of restaurant at a location
- **Book:** to book a table at the restaurant

IV. DEFINITION OF SERVICE NAMING

A **service naming scheme** is a system for assigning names to services. At first thought such a system may be quite simple but it is not. Indeed, a naming scheme includes several notions that can be quite confusing as follows:

- A **naming policy** provides guidelines aimed at ensuring the appropriate naming. Briefly, it indicates the aims of the naming scheme and consists of criteria or requirements such as:
  - Should be uniquely identifiable
  - Should be readable and understandable by human being
- A **naming convention** is a convention for naming things. The intent is to allow useful information to be deduced from the names based on regularities. For instance, in manhattan streets are numbered, with east-west streets being called "streets" and north-south streets called "avenues". Well-chosen naming conventions aid the casual user in navigating larger structures.

  In computer programming, a naming convention is a set of rules for choosing the character sequence to be used for identifiers in source code and documentation. Reasons for using a naming convention (as opposed to allowing programmers to choose any character sequence) include the following:
  - To reduce the effort needed to read and understand source code
  - To enhance source code appearance (for example, by disallowing overly long names or abbreviations).

  Naming conventions will ease the application of computerized lexical analysis and processing and will further [7]:
  - Reduce lexical variance
  - Increase human term recognition velocity
  - Increase precision in the interpreted meaning conveyed by name
  - Increase precision and recall in term query results
  - Help automatic string matching
- A **nomenclature** refers to a set or system of names or terms including the rules used for forming the names as those used in a particular science or art, by an individual or community, etc. Nomenclature may refer to biology, astronomy, chemistry, metallurgy, etc. A nomenclature includes hence a name policy and a name convention.
- A **namespace** is an abstract container providing context for the items (names, or technical terms, or words) it holds and allowing disambiguation of items having the same name (residing in different namespaces).
- A **name-to-object association** is called a name binding [6]. A name binding is always defined relative to a naming context. A naming context is an object that contains a set of name bindings in which each name is unique. Different names can be bound to an object in the same or different contexts at the same time.
- A **name resolution** (also called name lookup) is the process to determine the object associated with a name in a given context. To bind a name is to create a name binding in a given context. A name is always resolved relative to a context – there are no absolute names.

V. SERVICE NAMING POLICY

The high-level requirements on the service naming schemes are as follows:

1. The service naming must allow fast service discovery: when the mobile device moves from one area to another one, it should be offered the possibility to discover new services quite quickly in order to replace the currently in use services before they disappear.
2. A service name shall uniquely identify a service: once a name is registered for a service in a particular domain, there will not be any similar name to that service unless if it's under the other administrative domain.
3. A service name shall be comprehensive and give an indication about the services' purposes and functions: from the service name anybody should be able to have a guess on what the service is offering.
4. A service name shall be sufficiently simple to allow a fast service matching: the service matching here means the result of comparison between the user requirements and the service availability, where it occurs after the services are discovered. So, to offer service continuity to a mobile user who is changing environment it is crucial to have fast service matching which again requires simple service names.
5. It shall be possible to assign several names to a service in different languages: in a global environment with multiple cultures and languages, it is necessary to support service names in different languages.
6. It shall be possible to introduce aliases for a service: the reason for having aliases is to provide for transparency [8]. By having aliases it will help to identify the service function and makes the name easier to promote and remember. It also facilitates transfer of the service to different infrastructure when ever required.
7. It shall be possible to use a short name for a service: as the service naming is used to uniquely identify and addressed the service, hence long name might be confusing and error prone. So it is possible to use a short name for a service. As an example explained in Unmanaged Internet Architecture (UIA) [9], the name *work-pc* can be used instead of using long globally unique names like *work-pc.nor.item.ntnu.no* as been used in DNS. However in order to do that the name should be identical in its own administrative domain.
8. It shall be possible and simple to introduce new services: as the services are uncountable in numbers and there can be new services introduced everyday, the services need hence to learn about each other dynamically and cooperatively so the service naming should allow for introducing new services and it should be simple.
9. The service name scheme should be distributed and should allow more service providers to introduce any new service: there will always be multiple service providers which introduce new services in an uncoordinated way and the service name scheme should be able to cope with this.
10. The service naming scheme should be flexible and interoperable with existing service naming schemes: since there will always exist other service naming it is desirable that the new service naming can interoperate with them.

VI. SERVICE NAMING FOR FUTURE MOBILE SERVICES

The proposed service naming scheme must fulfill the requirements specified in section V as follows:

1. The service naming must allow fast service discovery:
    - This means that the service name must be as short as possible. We propose to use the commonly known noun of a service as its name. For example, the service to find and book a restaurant is simply named "restaurant".
2. A service name shall uniquely identify a service:
    - In order to avoid name collision, each service name should uniquely identify a service. Unfortunately, a short name as specified in the previous bullet may not be sufficient to uniquely identify a service because a noun may have different meaning in different context. For example, the word pen has five meanings:
        - Pen – a writing implement with a point from which ink flows
        - Pen – an enclosure for confining livestock
        - Playpen, pen – a portable enclosure in which babies may be left to play
        - Penitentiary, pen – a correctional institution for those convicted of major crimes
        - Pen – female swan
    - Consequently, it must be possible to expand the short name to a more composite or complete name, which uniquely identifies the service. It is necessary to have hierarchical structure, which allows gradual navigation towards higher hierarchical level and stepwise narrowing of the identification of the service and removal ambiguity.
    - For example, if "pen" is not sufficient to uniquely identify the service, a higher level component name may be fetched and use in the service matching. For example, the higher level name component of pen which is "writing" can be used in the service matching. If it is still ambiguous, the higher

name component, namely “instrument” can also be fetched and used.

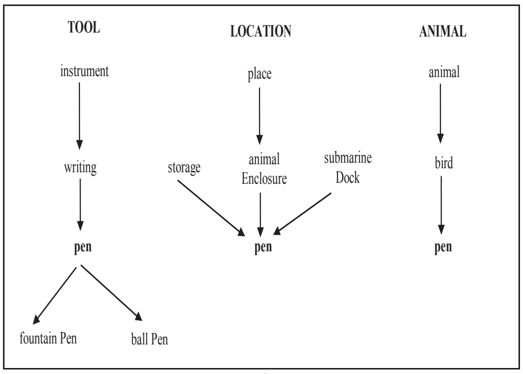

Figure 1. Example of the name structure hierarchy

- We propose that a service name structure which is inspired of the Uniform Resource Identifier (URI) as follows:
  - Semantically, the full name of a service will reflect the classification of the service in certain ontology. Figure 3 shows that the service “pen” may be belong to three different ontologies.
  - Syntactically, a name consists of several component names separated by period as follows:
    <componentnamen>..<componentname1>
    Ex: instrument.writing.pen
    The shortest name is always used first. If it is ambiguous the service name will be gradually expanded until the service is uniquely identified.
  - To name a service properly it is hence necessary to associate a service to ontology and to determine its upper class.

3. A service name shall be comprehensive and give an indication about the service’s purposes and functions:
   - Since almost everything can be a service from restaurant, cinema, hotel, taxi, etc. We propose to use the most common and shortest noun as name for a service. For example, there are many words which all denote a restaurant: restaurant, bistro, cafeteria, brasserie, etc. The most common word which is “restaurant” will then be used.
4. A service name shall be sufficiently simple to allow a fast service matching:
   - To allow fast service matching, the service name should not be too long but it must neither be too short and hence be confounded with other services. Such confusion will require a second round of name comparison which will be time consuming. With our service name scheme a service will have composite name that can be quite long. However, it may not be necessary to use the full name but only a partial name like “writing.pen”. It is a challenge to choose the optimal length for service names. We propose an empirical approach for the determination of the service name length in which the shortest name of a service with only one component name should be used first. When a need for further expansion in the service matching occurs the service name will update with a longer name with the number of component name corresponding the expansion required in the service matching.
5. It shall be possible to assign several names to a service in different languages:
   - A service name may have different names in different languages. For example, the service “taxi” is called “teksi” in malay, “taksiliito” in finnish, “cab” in uk, etc. These different names should belong to an equivalent class called taxi as follows:

     [taxi] = {x є S| X ~ taxi} S: set of services

     ~ is an equivalent relation which is
     - Reflexive: a ~ a
     - Symmetric: if a ~ b then b ~ a
     - Transitive: if a ~ b and b ~ c then a ~ c

     Alternative, the equivalent class taxi can be expressed as follows:
     [taxi] = {taxi, teksi, taksiliito, cab,..}
6. It shall be possible to introduce aliases for a service:
   - An alias is just another name that can be added to the equivalent class of the service as specified earlier.
7. It shall be possible and simple to introduce new services and multiple service providers should be able to introduce new services:
   - To introduce new service new name has to be created. To avoid name collision which can create confusion and increase the service matching time, it is necessary to check whether a name has been used before. If it is the case it may be actual to investigate whether it is the same service or not. If the services are similar there is no need to introduce a new one. If the services are different it may be necessary to examine the differences. If the differences are minor and the services are related to each other, it is necessary to classify the new service as either parent or child of the existing service class. If there are too many differences a

new service name should be chosen for the new service. One approach that can be used is by implementing a domain specific ontology where the same term may be used to denote different concepts. It is able to determine the relevance of an information source before accessing the underlying data and supporting wider accessibility of data via multiple ontologies and interoperation across them based on semantic relationships such as synonyms, hyponyms and hypernyms [10]. For name checking it is necessary to have a service registry which is ubiquitously available. For each service there should also be description or references to description that enable service providers to understand about the services. The service registry must also offer registration and deregistration service.

8. The service naming scheme should be flexible and interoperable with existing service naming schemes:
    - By allowing alias the proposed service naming scheme will allow associate a service class with existing services defined by existing systems such as Jini, UPnP, OMG, etc.

VII. CONCLUSION

Future mobile services should be discovered and composed dynamically according to the move of the user. In order to fulfill the real time requirement of the service discovery and matching it is crucial to have a sound and simple service naming scheme. In this paper, a service naming for future mobile services is proposed. This service naming allows full name which enable the unique identification of the services. However, to speed up the service matching only one or a few component names can be used for a service name. By introducing the service equivalent class the proposed service naming allows the usage of alias and names in many languages. The next step of this work will naturally be the implementation of the service naming scheme and the service registry. Since the 802.21 Media Independent Information Service is using XML and RDF (Resource Description Framework) it is natural that the proposed service naming scheme will be using the same technologies and eventually a more advanced semantic language such as the OWL (Web Ontology Language).

REFERENCES

[1] Media Independent Handover Services, "Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", IEEE P802.21/D7.1, August, 2007

[2] Kamal Bashah, N. S, Jorstad, I. & van Do, T, "Enabling Service Continuity on Future Mobile Services", Proceeding of the International Symposium on Wireless and Pervasive Computing (ISWPC2009), Feb 11-13 2009, Melbourne, Australia

[3] P. Mockapetris, "RFC 1035- Domain Names – Implementation and Specification", November 1987, online: http://www.ietf.org/rfc/rfc1035.txt

[4] RFC 2608, "Service Location Protocol", Version 2, online: http://www.ietf.org/rfc/rfc2608.txt

[5] Sun, "Technical White Paper: The Evolution of Jini Technology in Telematics", online: http://www.sun.com/software/jini, October 2001

[6] Object Management Group: Naming Service - Specification Version 1.3 Formal/04-10-03 - October 2004

[7] Daniel Schober: Naming Conventions for Ontology Engineering - The European Bioinformatics Institute (EBI) Net Project – Postdoctoral Ontologist – http://www.ebi.ac.uk/net-project

[8] Name Services, online: http://net.pku.edu.cn/~course/cs501/2005/ppt/chapter09__name_service_.pdf, 2005

[9] B. A Ford, "UIA: A Global Connectivity Architecture for Mobile Personal Devices", Ph.D. Thesis, Massachusetts Institute of Technology, September 2008

[10] Mena, E.; Kashyap, V.; Illarramendi, A. & Sheth, A. (1998), "Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure"

PAPER 3: Service Discovery in Future Open Mobile Environments

Nor Shahniza Kamal Bashah, Ivar Jørstad & Do van Thanh

Proceedings of 4th International Conference on Digital Society (ICDS2010), St.Maarten, Antilles, Netherlands, 8-15th February 2010

ISBN 978-0-7695-3953-9/10

Service Discovery in Future Open Mobile Environments

Nor Shahniza Kamal Bashah
Department of Telematics
Norwegian University of Science and Technology
O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway
nor@item.ntnu.no

Ivar Jørstad
Ubisafe AS
Gamleveien 252, NO-2624
Lillehammer, Norway
ivar@ubisafe.no

Do van Thanh
Department of Telematics
Norwegian University of Science and Technology – Telenor R&D
Snaroyveien 30, NO-1131 Fornebu, Norway
thanh-van.do@telenor.com

***Abstract*—Future mobile environment will see the open situation of mobile device that can be connected to several network systems offering heterogeneous sets of services using different names. In addition, the future mobile service can be anything and anybody including end-users can be service providers. This will result to the demand of having a dynamic and efficient service discovery and service matching. The paper discussed on the requirements of future service discovery based on the limitations found in the evaluation of existing service discovery schemes. Finally, the design of a future service discovery solution is proposed.**

Keywords-service discovery, service matching, dynamicity

I. Introduction

The advances in microelectronics and wireless access technologies allowing mobile terminal to be able to have many connections using different wireless access technologies with different networks. Consequently, the mobile terminal will be able to receive services from several heterogeneous networks. These can be different services or they can be different implementations of the same service which are competing for the selection of the user. Service discovery, which is the ability to discover and recognize the services available in a network will become one of the most important function for both the users and service providers. With the openness of future mobile Internet, it will allow anybody including the end-users to be service providers and a service can be anything. Hence it is unlikely that the service can have standardized name and type. For that, clarity and precision is a necessary condition guaranteeing that the service matching can always be accomplished in a determined period of time.

This paper starts with the study and evaluation of existing service discovery schemes followed by the identification of the high-level requirements of service discovery for future mobile services. Next the design of a future service discovery solution is elaborated and finally the paper ends with conclusions and propositions for future works.

II. Existing Service Discovery Schemes

In order to avoid re-inventing the wheel all the relevant existing service discovery schemes will be successively examined.

A. *Jini*

Jini is a distributed computing environment for entities running a Java Virtual Machine (VM) [1]. It has three main components called Services, Lookup Services and Clients. In order to discover the services and communicate with each other, the Clients and Service Providers need to locate the Lookup Service. There are three ways of doing it which are either by unicast, if the Clients know directly the address of the Lookup Service, multicast request from the Clients to the Lookup Services or multicast announcement from the Lookup Service to the Clients for the advertisement of the services available. Even though Jini has the code-mobility function which provides many benefits including non-protocol dependence but it appears in a discrete mobility environment where the availability of services are within the certain areas or access points only. The services are not available while moving from one area to another hence, might disrupt the seamless service connectivity.

B. *Universal Plug and Play (UPnP)*

UPnP is an architecture introduced by Microsoft for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices and PCs of all form factors [2]. It has three main components which are Control Points, Devices and Clients. Three steps involved in providing services which are to get an address, discovery and description. The description is divided into two which are a device description and one or more service descriptions; all are expressed in XML. After the three steps has accomplished, there are three alternative actions which are controlling the service, eventing the service and presenting the service which is in the HTML-based user interface. With the XML format used in its communication messages, it provides platform and programming language independent thus, greatly increases interoperability between devices. However, the devices in UPnP architecture need to be pre-defined thus limit the flexibility function of service discovery especially during the introduction of new services.

C. *Salutation*

Salutation is non-proprietary and royalty free service which intended for binding application, devices and services independent of OS, transport and product class [3]. It consists of Functional Unit (FU), Salutation Manager (SLM) and Transport Manager (TM). As a service broker, SLM provides four main functions which are service registry, service discovery, service availability and service session management. If there is a service available the FU will register with SLM which define the service function. When a client asks the SLM to provide on a particular service based on the requirement given, the SLM will reply to the request which give the information on the services that the FU offers. The interface between SLM and TM (called SLM-TI) makes Salutation as a processor, OS and communication protocol independent. Unfortunately the definition of the service should be accurate in order to find the match service. Otherwise client who searches for the specific service might get wrong return results hence, introduce long matching function or fail service discovery.

D. *Web services*

Web service (WS) is a software system designed to support interoperable machine-to-machine interaction over a network [4]. It consists of three main components: Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description Discovery and Integration (UDDI). In order to publish a service, the Service Provider needs to register with the UDDI first. The description and functions of the WS is written in WSDL. When the UDDI receives the service registration from the Service Provider, it may advertise or send announcement to other WS Clients on the existence and the availability of the services. The WS Client can also actively request for the service from the UDDI. If there's a service match, the UDDI will send the URL to the WS Client and the service can be invoked directly via the SOAP over HTTP. Even though the introduction of WS architecture has facilitated the business service information but there is a limitation in terms of the service discovery standards. Two identical service descriptions may mean very different things depending on the context of their use and proves to be a major limitation for capability matching.

E. *Limitation of current service discovery solutions*

Based on the evaluation of the existing service discovery schemes some major limitations are found as follows:

1) Since in the future mobile environment various network systems are competing to offer services to mobile users, the mobile terminal will then be able to receive services from several heterogeneous networks. The challenge now is to ensure that the services are still available and can be delivered to the user during the change of the network which is a major constraint in the existing service discovery schemes.

2) The service needs to be pre-defined (for example, in UPnP architecture the devices need to be pre-defined with vendor-specific information). Hence, does not support flexibility function especially when introducing new services.

3) The service name should be accurate in order to match with the user requirement or otherwise the service is not recognized (no service matching).

4) Different term and perspectives or knowledge of the service from the user and the service provider. For example, two identical services can have different name or the service with the same name can be totally or partially different.

III. FUTURE SERVICE DISCOVERY REQUIREMENTS

In future mobile environment mobile devices will be able to perform handover between access networks to improve connectivity as explained in [5]. To ensure service continuity, services in the network must be discovered rapidly. However, due to the plurality and diversity of the services introduced by different players at different times and places it is quite challenging to identify services rapidly and new service discovery system is required. By studying the future mobile environment where the mobile device can be connected simultaneously to several network systems offering heterogeneous sets of services using different names the following high-level requirements for future service discovery are derived as follows:

1. The service discovery must be efficient and capable to discover the services in a reasonable amount of time. Since the mobile devices may moves very rapidly from one network to another the service discovery must be able to find the new services very quickly in order to replace the currently in-use services before they are disconnected.
2. The service discovery must be interoperable and flexible especially during the introduction of new services. The

future service discovery must be able to discover services that are name according to current service discovery. Furthermore, in the future a service can be anything and offered by anybody. Consequently, the definition and naming of the service might be ad-hoc and the service discovery should allow introducing new services gradually.

3. The service discovery must be accurate and must not yield erroneous results. The future service discovery must not for example conclude that a service is doing one thing while it is offering something else.

4. The service discovery must be reliable and capable to discover all existing services. Even though the service name define might be different but the service discovery scheme should be able to recognize the service and know what they are doing.

5. The service discovery must be able to handle different terms used in the similar service. Some services might known or be called as other name in other countries or district. For example a flight service which is very common name in most part of the world, but in some countries flight or airplane is also known as other name such as “Fly” in Norway, “Avión” in Spain and “Kapal Terbang” in Malaysia. For that, the future service discovery should be able to understand the terms used and the services can still be discovered even though it is using different service name.

6. The service discovery must be able to handle that services with same name can be different. For example a user might think “Book service” as a service that provides information about books but service provider might use the term “Book service” as a booking service or reservation service.

7. The service discovery must be usable in a multi service provider environment where same service can be offered by several service providers with different names. Since there will always be multiple service providers want to introduce new service and it will happen in uncoordinated way, the service discovery should be able to allow the same service to be define in the equivalent services and from there different name can be introduced by several service providers.

8. The service discovery must be usable in a multi service provider environment where different services can be offered by several service providers with same names. There are also possibilities that the different services may be define with same names by several service providers, for example service provider A offering “Pizza service” as an order pizza service while service provider B might only offering “Pizza service” as a comparison of pizza price in the market. The service discovery should be able to distinguish between the services with the same name.

9. The service discovery must be able to handle services that are partially similar. Some services offered might look the same but can have partially different function, which we called as partially similar. For example, an “ordering service” can be a service offering from dialling a phone number to a party to make order until the service is delivered to the user but some “ordering service” might only provide the information of the contact person to make the order and the rest should be done by the user themselves.

10. The service discovery must be able to handle services that are a subset of another service. There are also some services that are subset of another service such as “Pay Taxi” which is a subset of a “Taxi” service. For that the service discovery should be able to define the subset service is in within the certain service domain ontology.

11. The service discovery must be updated and only available services shall be returned. Since in the future mobile service environment, the status of the services offered and its service information will be dynamically change thus it is very important to ensure that the service discovery been acquainted with the changes.

12. The service discovery must allow personalisation so that the service discovered matched with user preferences. Current trend of mobile services allow user to personalise the services that they subscribed or used based on their preferences.

13. The service discovery must be able to find services that are suitable to run with the specific device use by the user. Apart of knowing the capabilities of the services discovered, it is also important for the service discovery to have the knowledge about the user device capability (device-aware features) because some services that are discovered might not be compatibly matched with the device in use.

IV. DESIGN OF A FUTURE SERVICE DISCOVERY SCHEMES

Based on the limitations listed in Section II, the future service discovery requirements are deduced as explained in Section III. From there, the design process can be conducted. To have a clear and precise picture on the whole process of service discovery and service matching in future mobile services, we adopt the Unified Modeling Language (UML) [6] use case diagram illustrated to capture the high level requirements.

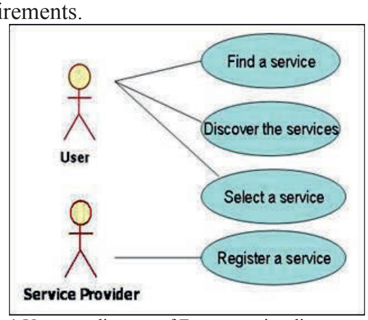

Figure 1 Use case diagram of Future service discovery system

As shown in Figure 1 there identified four use cases requested by two players, the user and the service provider as follows:

- User (client):
  - Discover the services
  - Select a service

    - Find a service
- Service provider:
    - Register a service

The four use cases represent the high level requirements of the future service discovery system.

A. *Use case "Register a service"*

In the current situation where the mobile phone is connected to a mobile network (e.g. GSM or UMTS), the service that is offered by the network has a standardized naming and service number, for example service name telephony with a dedicated service number which uniquely identify the service. From there if the client wants to find the service, a direct syntax matching can be done. But in the future, the mobile phone is connected to multiple heterogeneous networks and thus able to be access to services with different name that the client might not know. The issues are how the device can find out about the services and how the service is named. The syntax matching can no longer rely on the service name, service number, service type and attribute. To solve this issue, we propose a semantic service matching based on ontology. The convention on how a service name should be is clearly specified in the paper [7]. A template to be used by a Service Provider at registration of services is illustrated in Table 1.

Table 1 Example of Template for Future service naming scheme

Field	Description
Service name	A short 'noun' and common service name is used for e.g. "restaurant" to refer to place that serves food.
Service equivalent class	• It is used for service with different name but having same service instances. • It can also refer to services called by different languages (alias). For e.g. [Restaurant] = {Restaurant, Ristorante, Cafe, Warung, …}
Hierarchical structure	The hierarchical structure of the service name shows in which classification the service is in. This is very important especially for the service with the same name that can be different (e.g. Book that refers to an object or book which refers to a verb that means reservation)
Next level service	• If the service name is not sufficient during service matching, next level service name will be check until there is no ambiguity and the final service definition is retrieved. • It can also be used to find partial match (e.g. If the taxi service cannot be found, the service matching will go to the next level to see in which group/classification the service is in and suggest alternatives, in this case the bus or train which falls under the transportation group/classification is found as partial match)
Service name structure	The service name is in the URI format which contains <componentnamen>…<componentname1> For e.g. Reservation.Place.Book

This template will be written in XML and stored in the Service Infobase for Service Provider guideline. Figure 2 shows a collaboration diagram for a use case "Register a service" and Figure 3 is the sequence diagram for the use case showing the sequence of execution. As shown in Figure 2 the use case starts as follows:

1) When the Service Provider wants to register a service, it first needs to ask the Service Broker to check on the existing service name. If the service is already exist, the Service Provider will register the service with the same service name otherwise the Service Provider will need to look at the service equivalent to check on the same service with a different name. The Service Provider registers the service with service information reference (template shows in Table 1).

2) Once the Service Broker receives the registration, it will broadcast the service with service information reference to the Client and at the same time update the service info to the Service Infobase.

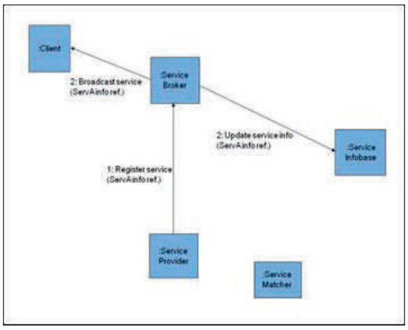

Figure 2 Collaboration diagram for "Register a service"

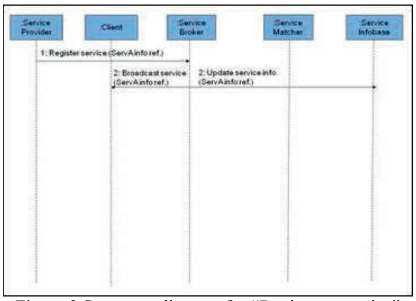

Figure 3 Sequence diagram for "Register a service"

B. *Use case "Discover the services"*

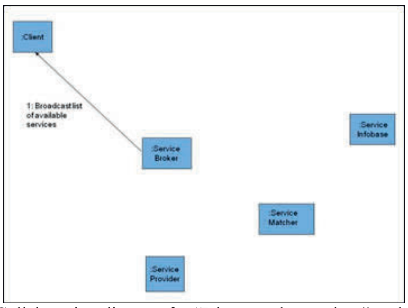

Figure 4 Collaboration diagram for "Discover the services" – alternative 1

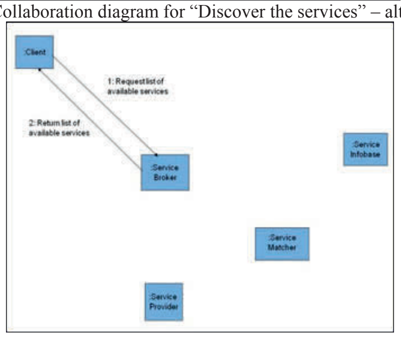

Figure 5 Collaboration diagram for "Discover the services" – alternative 2

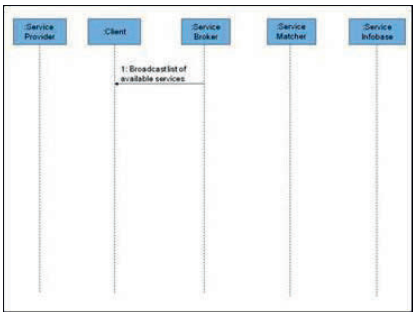

Figure 6 Sequence diagram for "Discover the services" – alternative 1

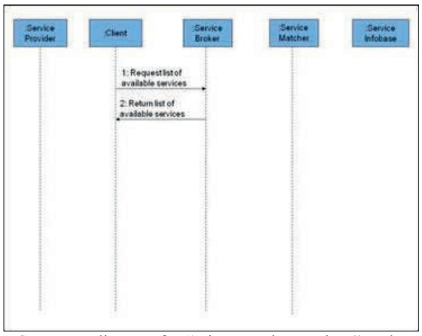

Figure 7 Sequence diagram for "Discover the services" – alternative 2

Figure 4 and Figure 5 shows collaboration diagrams for a use case "Discover the services" and Figure 6 and Figure 7 are the sequence diagrams for the use case with two alternatives, either the service broker is broadcasting the list of available services or the client request for the available services to the service broker. As shown in Figure 4 the use case starts as follows:

1) The Service Broker broadcasts the list of available services to the Client. There are two ways of broadcasting the services, either by periodically broadcasting the services within certain amount of time or upon new arrival/registration of a service. The broadcast method requires less power from the mobile phone since it can simply accept the list of available services from the service broker. However, the Client will need to synchronise before being able to get the list of available services and this may take time. The consequence is that the mobile device may loose contact with former service components before the new one is discovered and this result onto service discontinuity.

Figure 5 is the alternative for the use case "Discover the services" that starts as follows:

1) The Client first initiates a request on the list of available services. The request for the available services could be done in two ways: either the Client subscribes for the service or the Client just moves to a new area and would like to enquire on the available services.

2) Once the Service Broker receives the request, it will return the list of available services. For this method, the requester (Client) and the Service Brokers (there could be several service brokers at a location) are supposed to listen and respond to the request or in the case of Media Independent Handover (MIH) [8], there is a dedicated channel (or address) reserved for a unique and centralised Service Broker and the Client can send the request directly to it.

Therefore some important points need to be considered as follows:

- The list of available services must be usable and comprehensive but not too exhaustive due to the limitation in bandwidth of the wireless link, storage capacity of the mobile device, high latency, etc.

- The list should not contain all the service details but only the strict minimum such as service name and reference to service information.
- The list must have syntax and semantic which is comprehensive for both the client and the service broker.

C. Use case "Select a service"

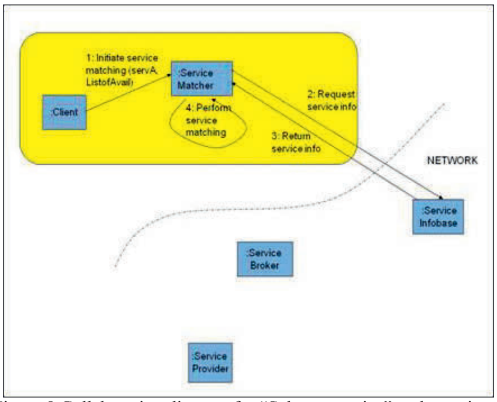

Figure 8 Collaboration diagram for "Select a service" – alternative 1

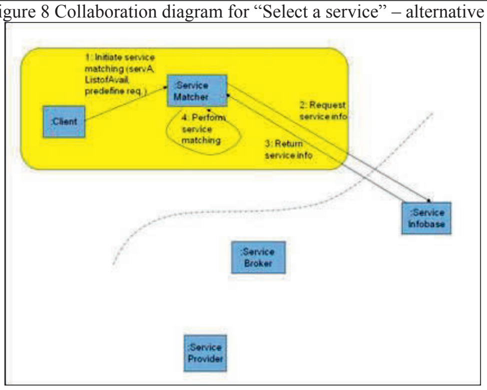

Figure 9 Collaboration diagram for "Select a service" – alternative 2

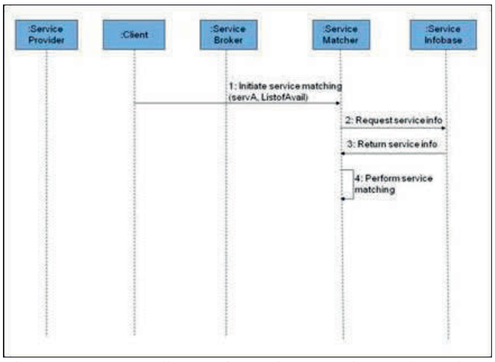

Figure 10 Sequence diagram for "Select a service" – alternative 1

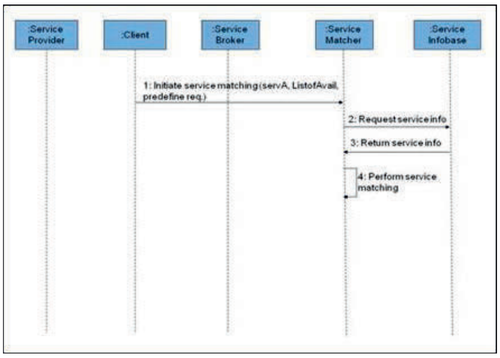

Figure 11 Sequence diagram for "Select a service" – alternative 2

Figure 8 and Figure 9 show the collaboration diagrams for a use case "Select a service" and Figure 10 and Figure 11 depict the sequence diagrams for the use case with two alternatives: either to have the selection of the service manually by the client or to automate the selection. As shown in Figure 8 the use case "Select a service" (manually) starts as follows:

1) The Client initiates service matching by sending the service name and the list of available services to the Service Matcher.
2) The Service Matcher then sends a request of service information to the Service Infobase.
3) The Service Infobase will return the service information to the Service Matcher.
4) Upon receiving the service information, the Service Matcher will perform service matching.

On the other hand, in the second alternative as illustrated in Figure 9, the selection of the service is done automatically and the use case starts as follows:

1) The Client initiates service matching by sending the service name, the list of available services together with predefine requirements (such as user personalisation and device adaptability) to the Service Matcher.
2) The Service Matcher then sends a request of service information to the Service Infobase.
3) The Service Infobase will return the service information to the Service Matcher.
4) After the service information is received, the Service Matcher will perform service matching and check also the predefined requirements sent by the Client. By implementing this method optimization will be achieved and results to an efficient service matching because in some cases there are possibilities of the matching result that does not comply/suit with user personalisation or user-device adaptability and is hence a waste of the time.

There are some important points that need to be considered as follows:

- It is necessary to introduce metrics for defining the 'best offer' for an automate service matching or via optimization algorithm implemented on the client's side, such as shortest routing path or lowest hop count.

- The Service Matcher could be located either in the Client side or in the Network side. The advantage of having the Service Matcher in the Client side is to ensure the mobile phone's full control on the service matching but this will consume more power. Having a service matching function in the Network side will reduce the processing time in the client and hence increase the performance efficiency but there is a risk for connection loss.
- The service information in the Service Infobase can be distributed in the World Wide Web (WWW) or stored in the MIH. If the Service Infobase is in WWW it will be accessible by any service providers. Being integrated in the MIH simplify the implementation

D. *Use case "Find a service"*

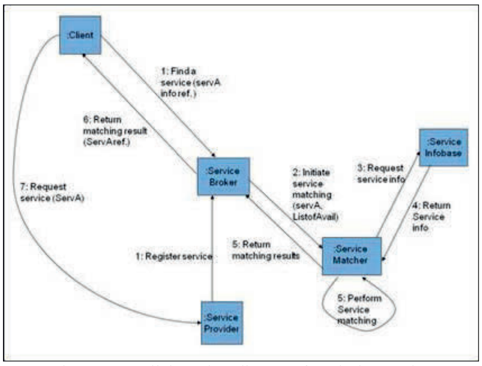

Figure 12 Collaboration diagram for Find a service

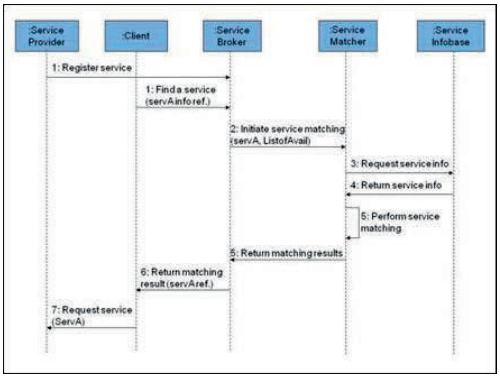

Figure 13 Sequence diagram for Find a service

Figure 12 shows collaboration diagram for a use case "Find a service" and Figure 13 depicts the sequence diagram. As shown in Figure 12 the use case starts as follows:

1) The Service Provider registers for a new service and when a Client moves to a location, it will find a service by sending service information reference to the Service Broker.

2) Service Broker will initiate service matching by sending the service name and the list of the available services to the Service Matcher. In this case, the Service Matcher is located at the network side.

3) The Service Matcher then send request of service information to the Service Infobase.

4) The Service Infobase will return the service information to the Service Matcher.

5) After receiving the service information, the Service Matcher will perform service matching and return the matching results to the Service Broker.

6) The Service Broker then will return the matching result together with service reference to the Client.

7) Upon receiving the service matched, the Client will send request for the service directly to the Service Provider.

CONCLUSION

The openness of future mobile Internet and the dynamicity of the mobile services has called for a sound service discovery and matching. The introduction of new service should be flexible and support for multi-service provider environment. In this paper, the design of a future service discovery solution is proposed. It has to also consider few situations which are illustrated by the use case diagrams. The next step of this work will be the implementation of the service discovery and matching. The visualization and implementation are also important tasks to be carried out.

REFERENCES

[1] Sun Microsystems. Jini^{TM} Architecture Specification Version 1.2. [Internet] December 2001. 10 July 2009. http://www-csag.ucsd.edu/teaching/cse291s03/Readings/jini1_2.pdf.

[2] Microsoft Corporation. Universal Plug and Play Device Architecture Version 1.0. [Internet] 15 October 2008. 5 August 2009. http://www.upnp.org/specs/arch/upnp-arch-devicearchitecture-v1.0.pdf.

[3] Salutation Consortium. Salutation Architecture Specification Version 2.0C-Part 1. [Internet] 1 June 1999. 23 July 2009. http://systems.cs.colorado.edu/grunwald/mobilecomputing/papers/salutation/sa20e1a21.pdf.

[4] W3C Working Group. Web Services Architecture. [Internet] 11 February 2004. 2 August 2009. http://www.w3.org/tr/ws-arch.

[5] Kamal Bashah, N.S, Jørstad, I. & van Do, T. *Enabling Service Continuity on Future Mobile Services,* Proc. of the International Symposium on Wireless and Pervasive Computing (ISWPC2009), Feb 11-13 2009, Melbourne, Australia. ISBN 978-1-4244-2965-3

[6] Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* Third Edition. Boston, MA: Addison-Wesley, 2004. ISBN 0321193687

[7] Kamal Bashah, N.S, Jørstad, I. & van Do, T. *Service Naming in Future Mobile Environments.*Proc. 38th International Conference on Parallel Processing (ICPP2009). September 22-25 2009. Vienna, Austria. ISBN 9780769538037

[8] Media Independent Handover Services. Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services. [Internet] D00.05, January

2006. 5 August 2009. http://www.ieee802.org/21/doctree/2006_Meeting_Docs/2006-03_meeting_docs/P802-21-D00-05.pdf

PAPER 4: An Open and Extensible Service Discovery for Ubiquitous Communications Systems

Nor Shahniza Kamal Bashah, Ivar Jørstad & Do van Thanh

Networked Services and Applications – Engineering, Control and Management (EUNICE 2010), Trondheim, Norway, 28 – 30th June 2010, LNCS 6164, pp. 272-273, 2010

ISBN-13 978-3-642-13970-3

An open and extensible Service Discovery for Ubiquitous Communication Systems

Nor Shahniza Kamal Bashah [1] Ivar Jørstad[2] and Do van Thanh[3]

[1] Norwegian University of Science and Technology, Department of Telematics, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway - nor@item.ntnu.no
[2] Ubisafe, Gamleveien 252, 2624 Lillehammer, Norway – ivar@ubisafe.no
[3] Telenor & NTNU, Snaroyveien 30, 1331 Fornebu, Norway – thanh-van.do@telenor.com

Abstract: In future ubiquitous communication systems, a service can be anything and introduced by anybody. Consequently, same or equivalent services may have different names and services with same name or type may be completely different. Existing service discovery systems are incapable of handling these situations. We propose a future service discovery, which is able to discover all these new service types. In addition, it is capable to find services that are not exact matches of the requested ones. More semantics are introduced through attributes like EquivalenceClass, ParentType and Keywords.

Keywords: Service advertisement, service discovery, service lookup, service matching, service registration

1 Introduction

In this research, a future service discovery system is proposed to solve the problem of discovering services in a ubiquitous communication environment, where similar services can have different names in different languages and services with same name may not offer the same functions and capabilities. The requirements on future service discovery systems are derived and more details can be found in [1]. The future service discovery system consists of service registration, service advertisement and request and service matching.

For the **service registration** the service provider must supply the details about the service that are necessary to detect, identify and use the service as follows:

- **Service name** – to denote and recognise the service. We propose to extend the service name format of future service discovery to allow any language.
- **Service type** – to carry on a search and identification of the functions offered by the service. There are three options of service type which are:
  - *Brand-new service type* – without relation with any existing service type. Hence the ParentType has to be set to nil and the EquivalentClass is then left empty.
  - *Equivalent service type* - the service type has a different name and may be another implementation but is equivalent to an existing service type. The service provider has to add the name of the existing service type and also all the known service type in different languages.
  - *Subtype of an existing service type* - The service type has all the functions and features of an existing service type but has also additional ones. The ParentType is hence indicated.

There are two alternatives for implementing **service advertisement**:

- The service registry or broker can take the initiative and broadcast the list of available services periodically.
- A client wanting to find services issues a request for a list of available services to the service registry.

However, both alternatives have their own advantages and disadvantages hence, selection of appropriate alternatives has to be considered and empirical methods have to be used. Anterior service advertisement may be a solution to be considered. Service advertisement can be made before the mobile user enters the location thus contribute to speeding up the service discovery and ensuring service continuity.

Instead of asking for the list of available services and carrying the service matching itself the client can **request** for a particular service or set of services. However, they may not have the opportunity to discover new and unknown services.

The **service matching** can be performed by the mobile client after the acquisition of the list of available services or by the network system upon receipt of a service request. We propose to extend the match operation on the Equivalence class, Service subtyping and Keywords of the requested service type.

2 Conclusions

This research explains the necessity of an innovative service discovery capable of handling similar services with different name and different services with same name. A prototype is currently under development and will be accomplished in the near future. Next, the experiments with various number of clients, number of services, different bandwidths and complicated service ontology will be carry out. The collected results will then be used to optimise the service discovery prototype. Another relevant and exciting future work will be the contribution to the extension of the 802.21 MIHF (Media Independent Handover Function) [2] for devices equipped with multiple wireless access technologies. Such an extension will ensure service continuity across multiple heterogeneous access networks.

References

1. Kamal Bashah, N.S, Jørstad, I. & van Do, T: Service Discovery in Future Open Mobile Environments - Proc. the Fourth International Conference on Digital Society (ICDS2010) - February 10-16 2010. St. Maarten, Netherlands Antilles (2010), ISBN: 9780769539539
2. Media Independent Handover Services, “Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services”, IEEE P802.21/D7.1, (August 2007)

PAPER 5: Service Discovery for Mobile Multi-domain Multi-language Environments

Nor Shahniza Kamal Bashah, Atif Bhatti, Imran Aslam Choudhary, Ivar Jørstad & Do van Thanh

Proceedings of 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010), Niagara Falls, Canada, 11 – 13th October 2010

ISBN 978-1-4244-7742-5/10

Service Discovery for mobile multi-domain multi-language environments

Nor Shahniza Kamal Bashah[1], Atif Bhatti, [2] Imran Aslam Choudhary[3], Ivar Jørstad[4] & Do van Thanh[5]

[1]Department of Telematics, NTNU, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway- nor@item.ntnu.no
[2]Department of Telematics, NTNU, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway- bhattiatif@gmail.com
[3]Department of Telematics, NTNU, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway- iachouhdary@gmail.com
[4]Ubisafe, Gamleveien 252, 2624 Lillehammer – Norway
[5]Telenor & NTNU, Snaroyveien 30, 1331 Fornebu, Norway

Abstract—**In mobile multi-domain multi-language environments, a service can be anything and introduced by anybody. Consequently, same or equivalent services may have different names and services with same name or type may be completely different. Existing service discovery systems are incapable of handling these situations. We propose a service discovery, which is able to discover all these new service types. In addition, it is capable to find services that are not exact matches of the requested ones. More semantics are introduced through attributes like EquivalenceClass, ParentType and Keywords.**

Keywords-service discovery; service lookup; service advertisement; service request; service matching; semantic matching

I. INTRODUCTION

The popularity of the mobile phones reflects the user's appreciation of the freedom, i.e. having access to any service anytime anywhere. However, so far the mobile services are mostly limited to voice communication and SMS (Short Message Service). Advances in wireless technologies have allowed mobile phones to be connected simultaneously to several network systems, e.g. GSM, GPRS, UMTS, WLAN, WiMAX, LTE, etc. and paved the way for innovative mobile services. But the success criterion is the existence of a sound, flexible and efficient service discovery which enables the clients to find and use the services. The future service discovery must be capable of finding relevant services offered by heterogeneous network systems. In such a ubiquitous communication environment, similar services can have different names in different languages. Furthermore, services with same name may not offer the same functions and capabilities. A future service discovery must be capable of dealing with the described challenges without confusion and returning correct answers in acceptable amount of time. In addition interoperability with existing service discovery systems must be ensured.

The goal of this paper is to present a service discovery system for mobile multi-domain multi-language environments. The paper starts with a brief review of the state-of-the art service discovery systems and their limitations. Next, is the specification of the requirements of the future service discovery system. The main part of the paper provides a comprehensive description of the system, which includes a service naming and description convention and an overall architecture. The sequence diagrams, system user interface, use scenarios and system performance results are given to illustrate the effectiveness of the proposed service discovery. Future works are summarized in the conclusion.

II. STATE OF THE ART SERVICE DISCOVERY

In this section the state of the art service discovery systems from both industrial research such as Jini and UPnP (Universal Plug and Play), and academic research such as SLS (Service Locating Service) are discussed.

A. Jini

In Jini, the service is classified by the types of the service object also known as service item. It contains three fields which are the `ServiceID` (globally unique 128-bit value generated by the Lookup service), `Service` (a reference to the object implementing the service) and `AttributeSets` (a set of tuples describing the service) [1]. The Client can request a service by using the `ServiceID` (if the Client knows exactly which service it wants). However, the `ServiceID` might sometimes be long and difficult to be remembered. Therefore a Lookup service is necessary. A Client locates an appropriate service by its type – that is, by its interface written in the Java programming language – along with descriptive attributes that are used in a user interface for the Lookup service. It is worth noting that two different service instances or two service types are not allowed to have same name. If the first level checking (first filter) does not give any result or it does give multiple results, the Lookup service will then check on the `AttributeSets` for the second level matching and to reduce the number of matching services. There can be multiple instances of the same class with different attribute values, as well as multiple instances of different classes in the service item contained in the service Lookup directory.

The Jini system uses the Java classes for a syntactic ontology (e.g. using printer class for a printer) and includes a

Lookup Attribute system to introduce more semantics. A wide variety of hierarchical views is obtained by aggregating items according to service type and attributes. However, it applies a strict naming convention which requires the uniqueness of the service name and type that cause inflexibility and does not suitable for future mobile environment which may have different services with the same name.

B. UPnP

The classification for UPnP is based on the device type which consists of a UUID (Universal Unique Identifier) and a URL (Uniform Resource Locator) of the description of the device [2]. This information is retrieved by the Control Point during the discovery. There are very few details about the device and a more detailed description of the device is then retrieved. The device description is divided into two logical parts: a device description describing the physical and logical containers and service description describing the capabilities exposed by the device. A single physical device may contain multiple logical devices which can be a single root device with embedded devices (and services) or multiple root devices (with no embedded devices). The UPnP Device Architecture defines a schema or template for creating device and service descriptions for any device or service type. Individual working committees subsequently standardize on various device and service types and create a template for each individual device or service type. Finally, a vendor fills in this template with information specific to the device or service, such as the device name, model number, manufacturer name and URL of the service description.

The service description in UPnP is at syntactic level hence the service matching is limited to syntactic comparison based on attributes or interfaces. It will not be able to detect a partially match such as the case where the service descriptions involve different representations of conceptually equivalent content. Furthermore, the device type must be predefined resulting to an inflexibility of naming and classifying of devices and services.

C. SLS

SLS [3], [4] is a prototype of a flexible service discovery system that implements dynamic tree structure for organizing SLS servers to meet the dynamic requirements of services and servers. The dynamic tree structure has the ability to quickly adapt to the changes imposed by creating, updating or deleting services and adding or removing SLS servers, and flexible as it has multiple service matching mechanism which contain an attribute matching engine and a semantic matching engine for different service interfaces. The service aggregation is apply for fast locating and improves efficiency as the lower-tiered SLS servers keep the detailed service information and the upper-tiered SLS servers aggregate the specific information into a more compact form. This aggregated information serve as routing index for a query to quickly find out a searching path that leads to the destination server.

The flexibility, scalability and support of dynamic changes in SLS features partially solved the future service discovery architecture issues. The services can be described and registered to a SLS server in various forms: a service template, a DAML description or Jini's interface. Hence, these multiple service matching mechanisms provide users a flexible means to discover mobile services. Unfortunately it will not be able to find services which are not exact matches with the requested one (partially match service).

III. FUTURE SERVICE DISCOVERY REQUIREMENTS

The requirements for future service discovery have been derived and highlighted in [5]. This section will specifically discuss thoroughly the requirements of future service discovery which are not fulfilled by the existing service discovery systems like Jini and UPnP.

The emergence of network technologies has allowed mobile user to have access in more than one network at a time. This results to the introduction of many services in different type of network systems. It is hence possible to conclude that:

"A service in the future multi-domain environment can be anything and be introduced by anybody at anytime."

This leads to a large variety of names and would result to the following situations:

1) *The same service may have different names and in different languages.*
   - Requirement: The future service discovery must be capable of handling the same service with different names in multi -languages.
2) *The same service name or word has several meanings and denotes different services.*
   - Requirement: The future service discovery must be capable of handling different services with same name without confusion.
3) *The service found may not be an exact match of the requested service but is a superset i.e. have extra functions in addition to the requested ones.*
   - Requirement: The future service discovery must be capable of discovering the services which are partially matches of the requested one.

IV. FUTURE SERVICE DISCOVERY SYSTEM

In this section the overall architecture of the proposed future service discovery and its development environment is described. The call flow on each method used is also explained in the sequence diagrams.

A. Service naming and description convention

According to the requirements described in the previous section, the service name and service type should not be restricted to any length or format. The service provider must have the freedom to the name and type as pleased. This requirement poses problem for the service matching to succeed.

To solve the problem we propose to introduce more semantics in the service description as in Figure 1. In addition to regular parameters like Name, Type, State Variable and Actions which provide the syntaxes of the service additional parameters like Keywords, ParentType, EquivalenceClass are introduced to provide more semantics to the service. They will be used in the service matching to find the requested services in an efficient and unambiguous manner. Their usage will be elucidated by the use scenarios in later section.

Service type description template

A service type description has the following items:

- *Name:* The name can be in any language and less than 64 characters.
- *Keywords:* Some words that can be used in the first round discovery
- *ParentType:* The name of service type that the current type is derived from.
- *ParentTypeURI:* The URI of the parent type
- *EquivalenceClass:* All alias in any language are given here
- State*Variables.* The *state variables* determine the states of the services. They are left empty in the service type description
- *Actions*: Actions are the methods that can be called by clients or other services
    - Each action has a *name* and a set of *parameters*
        - Each parameter has a *type*, *allowable values* (for enumerated types), and *direction* (in or out)
- *Events:* Enable clients to subscribe to the occurrence of a particular event

Figure 1. Service description template

B. Overview of the system architecture

Figure 2 illustrates the system architecture for future service discovery.

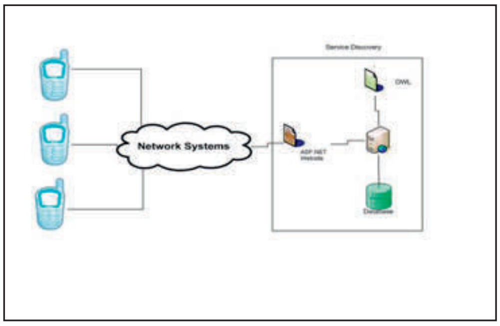

Figure 2. Future service discovery system architecture

The major components of the architecture are:

1) *Web Service Client* – This component is an application running on the remote mobile terminal. The Client on behalf of the user can ask/search for a service. It can also on behalf of the Service Provider introduce and register a new service. The protocol used between the Client and the actual XML Web Service is SOAP (Simple Object Access Protocol) [6].
2) Components of the service discovery system consisting of:

    i) *ASP .Net [7] Website (XML Web Service)* – This service acts as an interface between the Client/Service Provider and the System Server. The IIS Internet Information Services 5.01 or later is customized in order to publish the ASP.Net pages on the Internet.
    ii) *Database* – This component acts as a service repository which stores the services registered by the service provider.
    iii) OWL [8] file – is used for semantic matching with the OWL Lite (EquivalenceClass and ParentType).

C. Service discovery testbed

The hardware as well as the software components used for the system implementation are described below:

1) *Hardware* - A computer with installed Windows XP is used as a server. The server can process several requests from multiple clients in parallel. The Client can request the server by using any platform or browser to retrieve the desired services.
2) *Software*

    Operating System

    a) *Server side:* Since the application is developed using .Net technology, the .Net framework version 2.0 is installed on the server. Microsoft Windows based operating system is the minimum requirement for this version of .Net framework. Therefore Microsoft Windows XP was installed on the server.
    b) *Client:* The Client is independent of any architecture and hence can access the services by using any industry standard Internet browser from any operating system that has access to the Internet.
    c) *Programming languages:* For the implementation of this project Microsoft Visual Studio 2005 IDE for C# was used. The application was developed in C# whereas the website was developed using ASP.Net technology.
        - *C#:* C# was mainly used as an implementation language because of its convenient programming capabilities, the object oriented paradigm it supports, type-safety and wide range of libraries available

as a part of .Net framework [9]. Another reason to use C# was the OwlDotNetApi that was mainly used to access OWL file in order to define parent as well as equivalent classes for service types. The detail of this API can be found on [10].

- *ASP.Net:* ASP.Net is used for the XML web service and the web form interface for accessing the web service. Reason for using ASP.Net:
  - for building XML web service because it makes exposing and calling web services very simple.
  - for Client website because of the flexibility of interface it provides and support for the mobile devices. By building a website in ASP.Net one has to write the code once and the ASP.Net automatically generates pages based on the device they are called.

d) Support for ontologies: As the semantic meaning in this project of service discovery is being achieved by ontologies they were continuously monitored manually by using Protégé 4.0.2 [11]. OWL Lite is chosen in the implementation of this project due to its simplicity and the dynamic nature of the project even though it is known that it has some limitations as to compare with the OWL DL and OWL Full. Since the services can be introduced by anybody and there is no control mechanism on introducing the services, therefore it is required to avoid complexity in the system usage and for that only the EquivalenceClass and the ParentType properties is used which can be accomplished by the OWL Lite.

D. *Sequence Diagram*

The call sequence diagrams [12] are used to explain the flow of the four main methods in the future service discovery.

1) *Registration a service*: This method is used for the Service Provider to register a new service as illustrated in Figure 3.

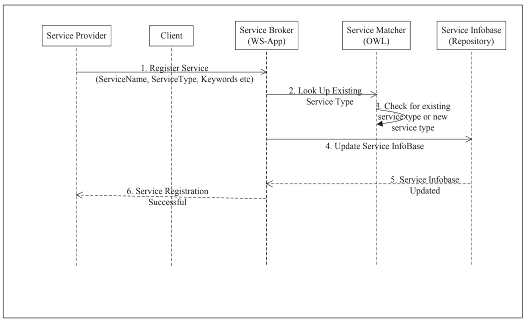

Figure 3. Call sequence diagram for Registration a service

i) When a Service Provider wants to register a service, he/she will first type the particular details of the service (e.g. the Service Name, Service Type and Keyword).

ii) The Service Broker which is the Web Service Application will then lookup for Equivalence Service Type.

iii) The Service Matcher which in this case is the OWL Lite will check for existing Service Type or new Service Type.

iv) The Service Broker will then update the Service Infobase which is the Service Repository of this system.

v) After that the Service Infobase is updated.

vi) The service registration is successful and Service Provider can see the new service registered in the service list.

2) *Discovering services*: This method is used for the Client which wants to find all available services as illustrated in Figure 4.

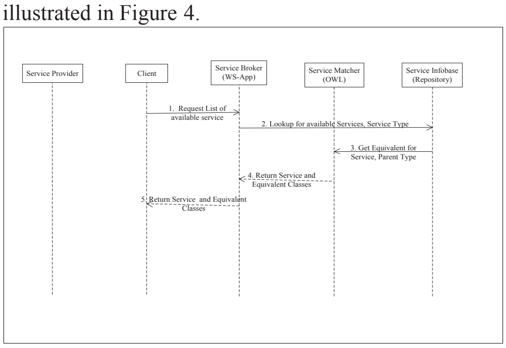

Figure 4. Call sequence diagram for Discovering services

i) When the Client wants to discover available services, it may request for list of services to the

ii) Service Broker.

iii) The Service Broker then will lookup for available services from the Service Infobase.

iv) The Service Infobase will then get the Equivalent for Service and Parent Type from the Service Matcher (OWL).

v) The Service Matcher will return the available services and all Equivalent Classes to the Service Broker.

vi) The Service Broker will return the services available and its Equivalent Classes to the Client.

3) *Service Request*: This method is used if the Client wants to find a particular service or set of services as illustrated in Figure 5.

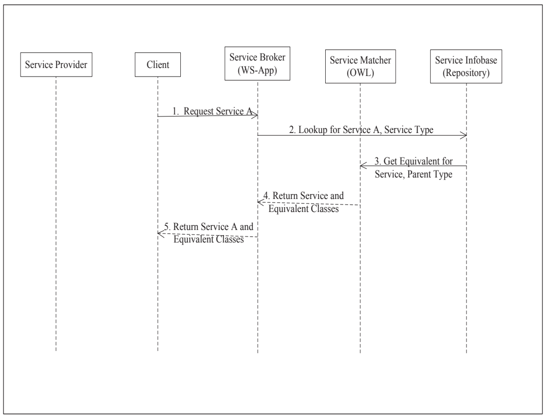

Figure 5. Call sequence diagram for Service Request

i) When the Client wants to request for a particular service, he may find the service either by using the Service Name or the Service Type.

ii) The Service Broker will Lookup for a particular service A and its Service Type in the Service Infobase.

iii) The Service Infobase will then get the Equivalent for Service A and its Parent Type from the Service Matcher (OWL).

iv) The Service Matcher will return the requested Service A and its Equivalent Classes to the Service Broker.

v) The Service Broker will return the Service A and its Equivalent Classes to the Client.

4) Find a service: This method is used if the Client wants to find a service but knowing very little information about the service (for e.g. just having a Keyword) as illustrated in Figure 6.

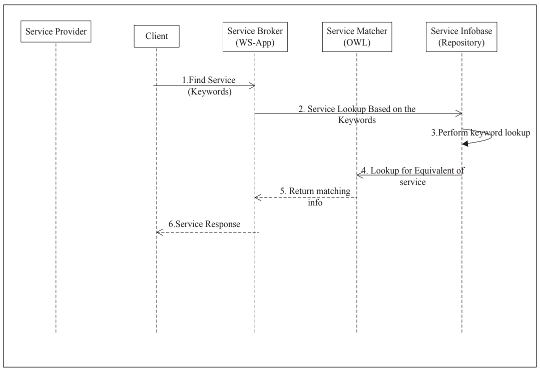

Figure 6. Call sequence diagram for Find a service

i) The Client can also find a service via the Keyword if he is not sure about the service Name.

ii) The Service Broker will do the Service Lookup based on the Keyword and send it to the Service Infobase.

iii) The Service Infobase will then perform the Keyword lookup.

iv) The Service Matcher (OWL) will then Lookup for Equivalent services.

v) Service Matcher will return the matching information to the Service Broker.

vi) The Service Broker will finally send the service response to the Client.

V. SYSTEM USER INTERFACE AND USE SCENARIOS

In this section the future service discovery system user interface is explained and based on the requirements specified in Section III, three use scenarios are selected for described in details.

A. System user interface

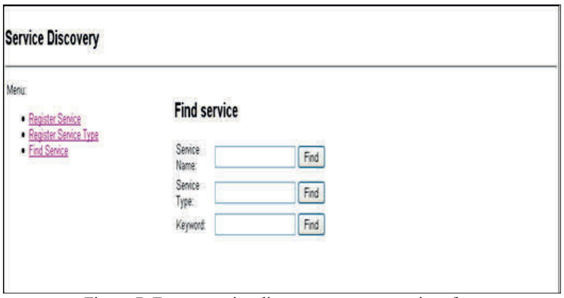

Figure 7. Future service discovery system user interface

Figure7 depicts the user interface for future service discovery system. It has three main methods which are:

i) *Register Service* – for the user to register a new service. The fields such as Service Name, Service URI, Service Description, Keyword, Service Type and Available Services are required to be filled.

ii) *Register Service Type* – during registration of a new service, if the user found that the service type of the registered service is not yet available, he may needs to register a new service.

iii) *Find Service* – there are three options to find service for the user. For finding a service by service name, the EquivalenceClass semantic will be used and all the similar services with different names and languages will be return. If the user chooses to find service by the service type, all the subtype services under the same ParentType will be return and if the

Keyword is used to find service, even the non-exact match services will be return as long as it has something in common to describe about the service (for e.g. Snacks or Cocktail to define Food service).

B. Use scenarios

Use scenario 1

The network system has similar services with different names and different languages.

When a Client wants to find a service, such as - a *Restaurant* service he will use either a standard English or his native language. The *Restaurant* service can be registered under other different names in multiple languages for e.g.

Restaurant: {Restoran, Ristorante, Café, Bistro, Warung, … }

If the Client enters the word *Café*, the future service discovery will return not only the *Café* service but also other similar service with *Café* in other names as depicted in Figure 8.

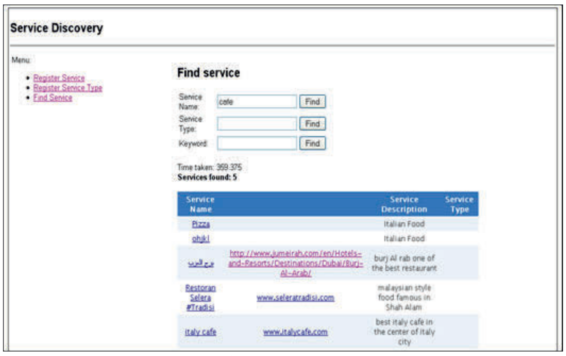

Figure 8. Use scenario 1a) - having similar service with different names

In addition, the future service discovery also supports for finding a service in multiple-languages as shown in Figure 9.

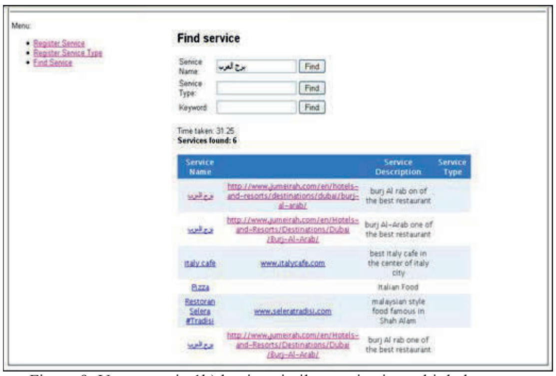

Figure 9. Use scenario 1b) having similar service in multiple languages

By having this feature the Client will not have to worry even though other names returned when he wants to find a service (especially if he is abroad and different languages is used to call a service) because the service discovery system returns only services equivalent to the one requested.

There are three options for the Client when it wants to find a service:

i) *by Service Name* – if the Client knows exactly the service it wants to look for

ii) *by Service Type* – if the Client is not sure about the Service Name but only knows the Service Type

iii) *by Keyword* – if the Client only knows very little about the service and has some clues describing the service (for e.g. Dining or Meal which refer to Restaurant)

Use scenario 2

The network system has different services with the same name.

Since in the future ubiquitous communication systems the service can be anything and introduced by anybody, there can be a possibility of having different term and perspectives of knowledge of the service from the user and service provider. The future service discovery supports the function of having different services with the same name as depicted in Figure10.

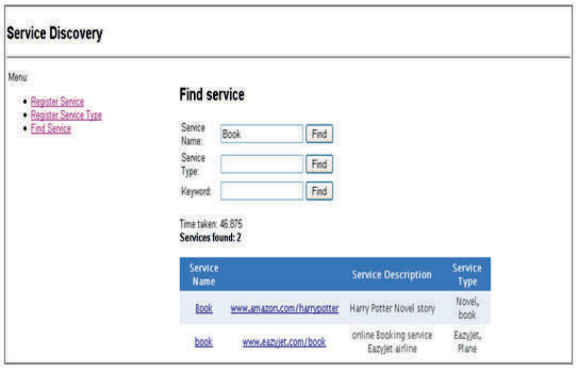

Figure 10. Use scenario 2 – having different services with the same name

As in the above example, the word Book can refer to a type of *Book* service (for e.g. buying online book or information about a book). There can also be other meaning of Book service which may refer to a Reservation service.

Even though the future service discovery allows having different services with the same name, the ambiguity and confusion can still be avoided via the details of the Service Description and Service Type returned during the service search.

Use scenario 3

The service discovery returns partially matches to requested service.

The future service discovery is using a semantic matching in addition to syntactical matching as the one used in the existing service discovery. This is very important especially in the situation where no equivalent service available but there exists service, which have additional functions to the requested ones. Such a service is called – partially match.

The future service discovery introduces the use of service subtyping by having ParentType and attributes ParentType in the Service Type description template. As illustrated in Figure 11, when the Client is asking for Telephony service; Skype, SIP and G1 are returned because they are grand children of Telephony (Skype and SIP are subtype of IP Telephony while G1 is subtype of GSM) and have inherited all the characteristics of Telephony. In this case the Skype, SIP and G1 are having similar functions (voice call – which is the generic Telephony features) but some of them have more or different additional functions (for e.g. Skype has video call feature) and they are also different in service components and service implementations.

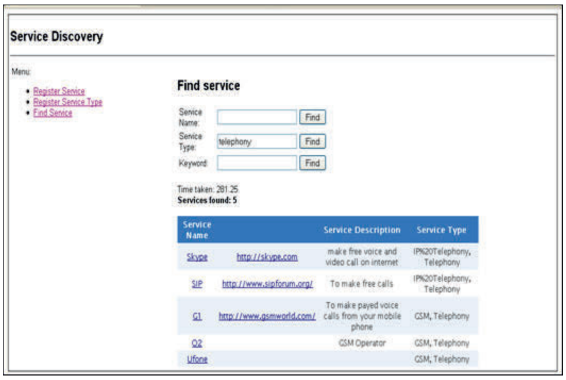

Figure 11. Use scenario 3a) – having partially match service

However, if the Client is asking for a specific Skype service, only the Skype is returned and nothing else. This is as illustrated in Figure 12.

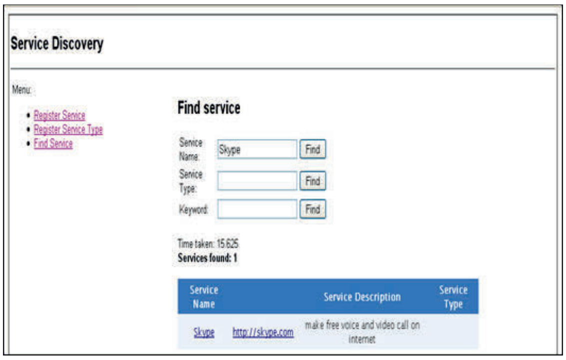

Figure 12. Use scenario 3b) – service subtype

VI. SYSTEM PERFORMANCE

As in the future mobile environments the performance of service discovery in dynamic changes is a major concern of the user. It is very important to ensure that the service requested is returned in an acceptable time (fast service discovery). Hence this section will show the result of our future service discovery system testing.

Figure13 shows the result of the testing which is done based on the four different search criterias:

i) find all available service
ii) find by service name
iii) find by service type
iv) find by keyword

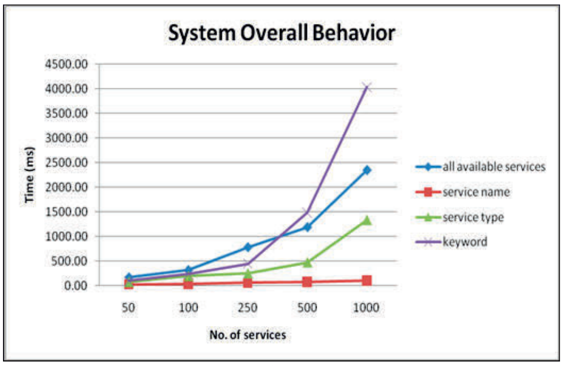

Figure 13. Future service discovery system performance

The x-axis shows the number of services used for testing while the y-axis is the time taken (in milliseconds) to return the results based on different types of parameters used in discovery of services.

The mean reading value for each number of service tested is used to plot the graph hence this result is retrieved. Service name lookup was really fast and scales well to real time tests that were conducted at different intervals by adding more services in the system repository. However the service lookup and key word look were considerably slower then preceding service name lookup as after retrieving the information from the database, lookup was carried out in ontologies to provide semantic meaning to the search and service discovery, hence making it gradually slower. Thus the system overall real time performance was found to be satisfactory and acceptable and it scales well to the increased number of services.

VII. CONCLUSION

In this paper a service discovery for mobile multi-domain and multi-language environments is proposed. To allow any service provider to introduce service as anything at anytime and anywhere more semantics are introduced. In addition to the service name and type, the relationship Keywords, ParentType and EquivalenceClass are introduced to concretize the service classification. A prototype is successfully implemented and is working fine. The next step will be to carry out larger experiments and tests on the system, e.g. increase the number of parents, the number of equivalent services, etc. It is also quite interesting to move the system to a real mobile and wireless environment where the client is implemented on real mobile phones.

REFERENCES

[1] Sun Microsystems. Jini™ Technology Core Platform Specification Version 2.0. [Internet] June 2003. http://www.scribd.com/doc/2280880/Jini-Technology-Core-Platform-Specification.

[2] Microsoft Corporation. Universal Plug and Play Device Architecture Version 1.0. [Internet] 15 October 2008. http://www.upnp.org/specs/arch/upnp-arch-devicearchitecture-v1.0.pdf.

[3] Tao Gu, Hung Keng Pung, and Jiankang Yao. Towards a Flexible Service Discovery. Elsevier Journal of Network and Computer Applications (JNCA), Vol. 28, Issue 3, pp. 233-248, May 2005. ISBN: 10848045.

[4] Tao Gu, Haichun Qian, Jianjang Yao, and Hung Keng Pung. An Architecture for Flexible Service Discovery in OCTOPUS. In Proc. of the 12th International Conference on Computer Communications and Networks (ICCCN '03), Dallas, Texas, October 2003. ISBN: 078037945403.

[5] Kamal Bashah, N.S, Jørstad, I. & van Do, T. Service Discovery in Future Open Mobile Environments. In Proc. of the 4th International Conference on Digital Society (ICDS2010), St. Maarten, Netherlands Antilles, February 10-16 2010. ISBN: 9780769539539.

[6] World Wide Web Consortium (W3C). SOAP Version 1.2 Part 1 – Messaging Framework (Second Edition). [Internet] 27 April 2007. http://www.w3.org/TR/soap12-part1/ .

[7] Onion, Fritz. "Essential ASP .Net with Examples in C#". Microsoft .Net Development Series. Boston, MA: Addison Wesley, 2004. ISBN 0201760401.

[8] World Wide Web Consortium (W3C). OWL Web Ontology Language Overview. [Internet] 10 February 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/#s3.

[9] Blum, Richard. "C#™ Network Programming". Alameda, CA: John Wiley, 2003. ISBN: 0782141765.

[10] OwlDotNetApi. [Internet] http://users.skynet.be/bpellens/OwlDotNetApi/documentation.html.

[11] Holger Knublauch, Ray W. Fergerson, Natalya F. Noy and Mark A. Musen. "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications". The Semantic Web (ISWC 2004), 2004, pp. 229-243.

[12] Fowler, Martin. "UML Distilled: A Brief Guide to the Standard Object Modeling Language". Third Edition. Boston, MA: Addison Wesley, 2004. ISBN 0321193687.

PAPER 6: Service Discovery in Ubiquitous Mobile Computing Environment

Nor Shahniza Kamal Bashah, Natalie Kryvinska & Do van Thanh

Proceedings of 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2010), Paris, France, 8 – 10th November 2010

ISBN 978-1-4503-0440-5

Is not included due to copyright

PAPER 7: A Mobile Service Architecture for improving Availability and Continuity

Nor Shahniza Kamal Bashah, Ivar Jørstad, Do van Thuan, Tore Jønvik & Do van Thanh

Proceedings of 2011 IEEE Symposium on Computers and Informatics (ICSI2011), Kuala Lumpur, Malaysia, 20th – 22nd March 2011

ISBN 978-1-61284-690-3

A Mobile Service Architecture for improving Availability and Continuity

Nor Shahniza Kamal Bashah[1], Ivar Jørstad,[2] Do van Thuan[3], Tore Jønvik[4] & Do van Thanh[5]

[1]Department of Telematics, NTNU, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway- nor@item.ntnu.no
[2]Ubisafe, Gamleveien 252, 2624 Lillehammer, Norway- ivar@ubisafe.no
[3]Linus AS, Norway- t.do@linus.no
[4]Oslo University College, Norway- tore.jonvik@iu.hio.no
[5]Telenor & NTNU, Snaroyveien 30, 1331 Fornebu, Norway

***Abstract*- In mobile multi-network multi-domain environments services can be anything and can have any name. This poses problems for the service availability and consequently service continuity since services may not be discovered in time for being used. Services offering equivalent functions and capabilities such as GSM voice, IP telephony, etc. are usually considered as different by the network system and are not interchangeable. This paper proposes a mobile service architecture which enables the discovery and consumption of equivalent services. The solution provides also seamless service continuity by incorporating dynamic discovery of equivalent services with dynamic switching of available services. The solution offers personalisation while preserving both flexibility and control of the user. Although most of the storage and processing of data are located in the network the service continuity process is initiated, controlled and decided by the mobile device's side. A proof-of-concept including a client implementation on Android is presented.**

Keywords: Service continuity, service availability, voice call continuity, service transfer, handover, service personalisation

I. INTRODUCTION

A modern mobile environment consists of multiple network systems own and managed by different players such as home, enterprise, airport, shopping center, etc. The services offered by these network systems are usually not standardised. Consequently, the same services may have different names and different services may have same name. This creates problems for service availability and continuity simply because services are not discovered in order to be used. For example, a user talking using GSM voice service will have problem to find an IP telephony called chIPhone offered by a shopping center in order to transfer his call over. In fact, even though the user perceives the GSM voice service and chIPhone as equivalent these two services are completely different in terms of implementation, user interface and ownership and there is no relationship between them. In order to improve the service availability and to provide service continuity across network systems this paper proposes a new mobile service architecture that enables the unification of equivalent services and hence enhances the service availability and continuity for the user. The paper starts with an explanation of the motivations of the proposed architecture. Next, the related works are briefly reviewed. The concept of service is then clarified before the main part of the paper, which is the elaboration of the service continuity architecture. The proof-of-concept implementation is described thoroughly. Further works are given in the conclusion.

II. MOTIVATION

A. Handling of equivalent services

In a multi-network environment a service can be implemented using different architecture and technologies for different network systems resulting to different user interfaces. For example, the telephony or voice service can be implemented as GSM voice or chIPhone using IP technologies. Seen from the user these two services are equivalent since they both offer human-to-human voice communication. However, they are currently considered as completely unrelated. When the mobile device is moving connections to the network systems are also changing, a service currently in use e.g. GSM voice may no longer be available and another one e.g. chIPhone may appear. Unfortunately, the mobile device is not aware of the availability of this equivalent service in order to use it. Of course, the detection and swapping to the pop up service is not only motivated by the availability but also other factors like ease of use, performance, security, price, etc. Furthermore, when a service e.g. voice service is ongoing it is desirable that it is not interrupt. Service continuity is another required characteristic. Seen from the user it is valuable to be able to manage this sort of equivalent services in a uniform and simple way to improve service availability and preserve service continuity.

B. Providing location-based services

When travelling the mobile user will want to have access to the same services such as taxi, hotel, restaurant, flight booking, etc. which enable the necessary booking, modification, cancellation, etc. when needed. At first glance, the mentioned services look the same everywhere but a closer look reveals the opposite. Indeed, when being in Trondheim the user will need to communicate with the Trondheim Taxi service and not the one in London. It is also desirable that the generic Taxi service switches to the right service according to the location automatically without the intervention of the user. In fact, it may be sometime difficult for the user to find the Taxi service in a country, e.g. China, Cambodia, etc. where a different language is used. Seen from the user it is valuable to seamlessly access the right service based on the location.

C. Offering personalised service

As human being each user will have some preferences related to the services and it will be very convenient if these preferences are remembered and dynamically re-applied to every equivalent service in use. In addition, the personal data such as personal details, address, phone numbers, contact list, etc. should also be shared between all the services. Seen from the user it is valuable to be able to use the service in a personalised way and personal data can be handled at one place.

III. RELATED WORKS

A. Service continuity works at NTNU

There have been works on service continuity related to the future mobile research activities carried out by Do and Jørstad [1] [2]. The following definitions have been proposed in [3]:

"Seamless service continuity is the ability to pick up a service at a new Location, where service abruptness is bounded by the time it takes the user to move between the two Locations."

"Non-seamless service continuity is the ability to pick up a service at a new Location, where service usage can proceed from the 'point'/state where it was left at the previous Location, but where additional disruption is introduced due to a required reorganization of the service composition."

Several interesting concepts and theoretic models have been developed but no real implementation or usage has been achieved.

B. TISPAN (Telecoms & Internet converged Services & Protocols for Advanced Networks)

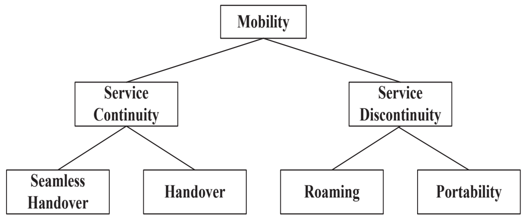

Fig. 1. TISPAN's mobility components

In their works of defining the European view of the Next Generation Networking (NGN), TISPAN has studied all the potential service capabilities and requirements [4] and defined the concept of service continuity as a component of mobility [5] as shown in Fig. 1.

More specifically, service continuity is defined as "*the ability for a user to maintain an ongoing service during mobility*"

Unfortunately, although the definition may imply continuity across different devices, it is not clearly stated.

C. 3GPP Voice Call Continuity (VCC) and GSMA VoLTE

The 3GPP has conducted studies described in TR 23.806 [6] and has elaborated in TS 23.2006 [7] the specifications of the Voice Call Continuity (VCC) which ensure the persistence of a voice call when the mobile phone moves between circuit switched and packet switched radio domains.

With the emergence of LTE (Long Term Evolution) paving the way for VoIP over all-IP mobile networks, it is necessary to ensure the continuity of voice calls when a user moves from an LTE coverage area to another without LTE. The GSM Association VoLTE (Voice over LTE) initiative endorses the IMS solution specified by 3GPP. VoLGA (over LTE Via Generic Access) is ensuring continuity by tunneling GSM voice over IP.

The focus of all these activities is the seamless switching between circuit switched voice and IMS voice, which are two well-defined services provided by mobile operators. Our work extends service continuity over non-standardised services.

D. IEEE Media Independent Handover (MIH)

The IEEE Working Group has introduced the 802.21 Media Independent Handover (MIH) Services [8] which allows the mobile terminal to detect the communication channels and services that are available in the vicinity. It facilitates the handover between heterogeneous networks and ensures seamless connectivity via the Media Independent Handover Function (MIHF) which is a logical entity in the mobility-management protocol stack of both the mobile node and the network elements.

With the 802.21, the Mobile Node can communicate with the visited MIH Information Server to acquire network information necessary for carrying out handover and preserving connectivity.

However, it can only support the service continuity if the same service implementation is used across all heterogeneous access networks. We extend the service continuity for future mobile service architecture where the services offered by different types of networks could be different in service implementation and using different components.

IV. Enhancing Availability And Providing Continuity

A. *Clarification of the service concept*

Service is a concept which is broadly used by numerous communities and hence quite confusing. To avoid confusion, the definition used in this paper is adopted from [9] as follows

"A service is a mechanism enabling the end-user's access to one or more capabilities"

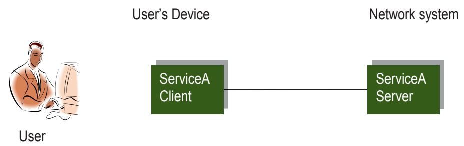

Fig. 2. A common service architecture

In a mobile environment, a service can be conceptually realised by two components:

- A **server** on the network, which can be both hardware and software and distributed over multiple network elements.
- A **client** on the user's device, which can be:
  - **Generic client** such as the Web browser that is capable to collaborate with any service server.
  - **Specific or dedicated client** that could collaborate only with some specific service servers.

In this paper, the focus is only on specific clients and how to provide service continuity on these clients.

B. *A Service Continuity architecture*

When the user is moving the service A as shown in Fig. 3 currently in use may cease to be available. However, a service B which is perceived as equivalent to service A by the user is emerging. The user will probably want to use service B instead of service A and the replacement should preferably be carried out in smooth and seamless way. Until now this is not possible since the two services are considered as completely different and there is no mechanism enabling the service switching.

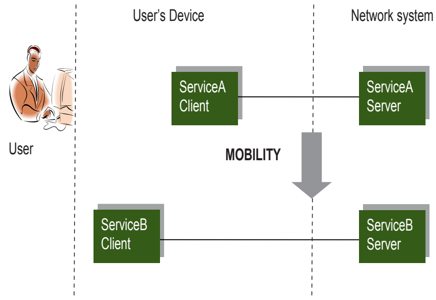

Fig. 3. Services in a mobile environment

To enhance the service availability and continuity we propose to introduce a new entity called **Generic Service** on the user's device. The Generic Service is equipped with a user interface that enables to receive commands from the user and to generate output. Depending on their availability the Generic Service will either invoke service A or service B.

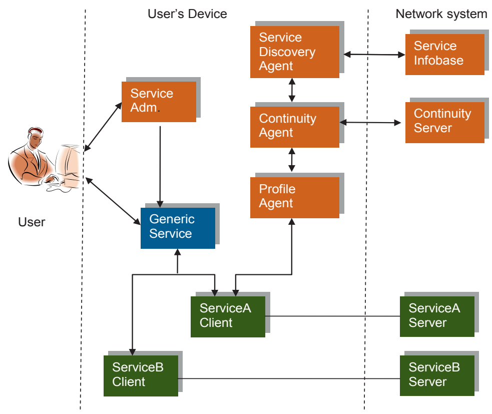

Fig. 4. A Service Continuity architecture

Now, the challenge is to know when and how to switch the Generic Service to either service A or service B. Another entity called **Continuity Agent** is proposed to assist the Generic Service in the switching process between services. It will perform the necessary initialisation and configuration such that the replacing service can continue from the point the former service stops. In the case where a new service client is required the Continuity Agent should also be capable of downloading, installing and initiating this new client. The Continuity Agent gets assistance from the **Continuity Server** on the network system to carry out seamless transfer of synchronous services like telephony.

The Continuity Manager needs assistance to find dynamically the available services at every place. An entity called **Service Discovery Agent** is introduced to perform the discovery of available services in collaboration with an entity called **Service Infobase** in the network system. The Service Infobase is a service registry that contains information about all the services available at a location. It has also capabilities to provide a complete list of available services to carry out matchmaking to find the specific services requested by the Service Discovery Agent. The service information stored in the Service Infobase should include details about service equivalence as specified in [10] which allows the discovery of equivalent services. The Service Infobase is also equipped with an open interface enabling all the service providers, including the end-users to register the services they are providing. The Service Infobase provides also an *Event service* that the Service Discovery Agent can subscribe to receive notification when networks and services emerge due to the user's mobility.

At service switching, to avoid cumbersome repetition it is convenient for the user if the preference settings and personal data can be shared among the equivalent services. For that the **Profile Agent** is introduced to store and administer the user profile. It has interface allowing the Continuity Agent to fetch the necessary data to configure the service and to save all the changes done by the user when using the services.

The whole service continuity process is illustrated in Fig. 5.

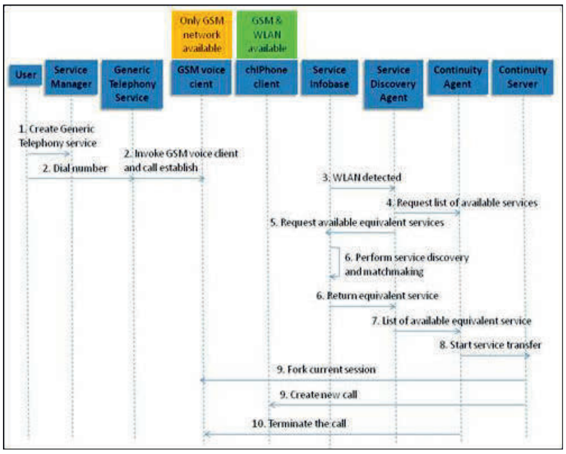

Fig. 5. Sequence diagram for the service continuity process

13. The user communicates with the *Service Manager* to create a *Generic Telephony service*, which includes either *GSM voice* service or *chIPhone*, a SIP-based IP telephony service. The GSM voice client is factory installed at while *chIPhone* is downloaded and installed by the user.
14. The user dials a number on the Generic Telephony service at a place where only GSM voice is available. The *Generic Telephony service* invokes the *GSM voice client* and the call is established.
15. The user is moving to a place where there is WLAN network offering an IP telephony service. The *Service Infobase* sends notification to the *Service Discovery Agent.*
16. The *Service Discovery Agent* requests the list of services demanding availability and continuity from the *Continuity Agent*, which contains *GSM voice* and *chIPhone.*
17. The *Service Discovery Agent* requests the *Service Infobase* to find all the available services that are equivalent to *GSM voice* and *chIPhone.*
18. The *Service Infobase* carries out the service discovery and matchmaking, and returns to the *Service Discovery Agent* a list containing a service *sIPhone*, which is SIP-based telephony and equivalent to *chIPhone*.
19. The *Service Discovery Agent* hands the list of available services over to the *Continuity Agent* for the initiation of the service transfer
20. Since the chIPhone client is already a SIP client, the *Continuity Agent* does not have to download and install any SIP client but only initialize the *chIPhone client*. The *Continuity Agent* requests the *Continuity Server* to start the service transfer.
21. The *Continuity Server* forks the current session between a B-party and the *GSM voice* client and creates a new call leg towards the *chIPhone client*.
22. When the *chIPhone client* is ready and running the Continuity Agent terminates the call leg with the *GSM voice client*. This is a typical soft handover since there is a short overlapping period where both clients are operating.

V. IMPLEMENTATION

A proof-of-concept prototype demonstrating the proposed service continuity architecture is under development.

The **Continuity Server** is realised by the Voice server inherited from the EUREKA Mobicome[1] project [11], which is focusing on providing service continuity across multiple devices. For example, the user is talking to the mobile phone while walking to his office. When arriving to his office, he transfers the call to his multimedia PC and continues the conversation while taking notes since he does not have to hold the phone anymore. However, it is worth noting that changing devices is conceptually similar to changing voice clients in the same device. The only difference is that in the first case the user is fully aware and actively participating to the switching of devices while in the latter case service continuity should preferably be seamless.

A simplified version of the **Service Infobase** supporting the discovery of equivalent services in different languages has been successfully implemented [12]. In addition, the

satisfying only a subset of the functionality has been

[1] The EUREKA Mobicome project (2007-2010) is aiming at providing fixed-mobile convergent IMS environment with the participation of Telenor, Telefonica, Ericsson, Huawei, Linus, Ubisafe, HiQ, Oslo University College, Polytechnical University of Madrid and Blekinge Institute of Technology

realised.

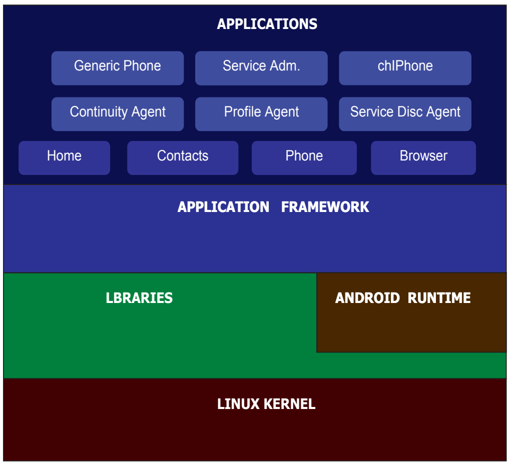

Fig. 6. A Service Continuity implementation on Android

The development of entities on the mobile device is carried out on Android [13], which is a software stack for mobile devices that includes an operating system, middleware and key applications. As shown is Fig. 6 the service continuity entities are introduced as applications. The *Generic Phone* will switch between the native *Phone* application and *chIPhone*, a SIP client. The *Profile Agent* will communicate with the native *Contacts* application to provide proper configuration and contacts for *Phone* and *chIPhone*.

VI. CONCLUSION

This paper demonstrates the needs for improving service availability and continuity in a heterogeneous networks environment where there are equivalent services with different names. To enable the use of these equivalent services it is proposed a service architecture, which incorporates dynamic discovery of equivalent services and dynamic switching of available services. The solution also ensures the re-use of the user's preference settings and personal data including personal details, contact, pictures, etc. Although most of the storage and processing of data are located in the network the service continuity process is initiated, controlled and decided by the mobile device's side. Currently, the only criterion for the transfer of service is the service availability. As further work it is quite relevant to introduce additional handover criteria such as QoS, security, price, etc. It might also be interesting to consider the implementation of the proposed service continuity for LTE (Long Term Evolution). In early phase LTE will only be deployed in limited areas and it is necessary to support seamless handover to 2G/3G network.

REFERENCES

[1] Ivar Jørstad, Schahram Dustdar and Do van Thanh, "Service Continuity and Personalisation in Future Mobile Services", Proceedings of the 10th Eunice Summer School, ISBN 952-15-1187-7 (pp. 128-134), Tampere, Finland, June 14-16, 2004.

[2] Ivar Jørstad, Do van Thanh and Schahram Dustdar, "An analysis of service continuity in mobile services", Proceedings of the 2nd International Workshop on Distributed and Mobile Collaboration (DMC 2004), IEEE Computer Society Press, ISBN 0-7695-2183-5 (pp. 121-126), University of Modena and Reggio Emilia, Italy, June 14-16, 2004.

[3] Ivar Jørstad, Do van Thanh and Schahram Dustdar, "A Service Continuity Layer for Mobile Services", Proceedings of the IEEE WCNC Wireless Communications and Networking Conference 2005, ISBN 0-7803-8967-0, New Orleans, USA, Mar 13-17, 2005.

[4] Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN): Services Capabilities, Requirements and strategic direction for NGN services – Technical Report – ETSI TR 181 003 V1.1.1 (2006 05).

[5] Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN): NGN Terminology - Technical Report – ETSI TR 180 000 V1.1.1 (2006 02).

[6] 3rd Generation Partnership Project: 3GPP Technical Specification Group Services and System Aspects; Voice Call Continuity between CS and IMS (Release 7) – Technical Report TR 23.806 V7.0.0 (2005-12).

[7] 3rd Generation Partnership Project: 3GPP Technical Specification Group Services and System Aspects; Voice Call Continuity (VCC) between Circuit Switched (CS) and IP Multimedia Subsystem (IMS); Stage 2 3GPP TS 23.206 V7.5.0 (2007-12).

[8] IEEE Computer Society, "IEEE Standard for Local and Metropolitan Area Networks. Part 21: Media Independent Handover Service", IEEE Std. 802.21TM – 2008, 21 January 2009.

[9] Nor Shahniza Kamal Bashah, Natalia Kryvinska & Do van Thanh, "Service Discovery in Ubiquitous Mobile Computing Environment", Proceedings of the 8th International Conference on Advances in Mobile Computing & Multimedia (MoMM2010), ISBN 978-1-4503-0440-5, Paris, Nov 8-10, 2010.

[10] Nor Shahniza Kamal Bashah, Ivar Jørstad & Do van Thanh, "Service Discovery in Future Open Mobile Environments", Proceedings of the 4th International Conference on Digital Society (ICDS 2010), IARIA, ISBN: 9780769539539, St. Maarten, Netherlands Antilles, February 10 – 16, 2010.

[11] Do van Thuan, Ivar Jørstad, Boning Feng, Tore Jønvik & Do van Thanh, "Service Continuity for IMS converged services", Proceedings of the 10th International Conference on Intelligence in Next Generation Networks (ICIN 2008), ISBN 978-1-60702-868-0.

[12] Nor Shahniza Kamal Bashah, Atif Bhatti, Imran Aslam Choudhary, Ivar Jørstad & Do van Thanh, "Service Discovery for mobile multi-domain multi-language environments", Proceedings of the 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMOB 2010), ISBN 978-4244-7741-8, Niagara Falls, Canada, October 11-13, 2010.

[13] Android developers – http://developer.android.com/index.html

PAPER 8: Service Continuity across heterogeneous networks

Nor Shahniza Kamal Bashah, Natalia Kryvinska & Do van Thanh

International Conference on Software and Information Engineering (ICSIE 2011), Kuala Lumpur, Malaysia, 17th – 19th June 2011

ISBN 978-0-7918-5973-5

Is not included due to copyright

PAPER 9: Novel Service Discovery Techniques for Open Mobile Environments and their Business Applications

Nor Shahniza Kamal Bashah, Natalia Kryvinska & Do van Thanh

Third International Conference on Exploring Service Sciences (IESS 2012), Geneva, Switzerland, 15th – 17th February 2012, LNBIP 103, pp. 186-200, 2012

ISBN 978-3-642-28226-3

Novel Service Discovery Techniques for Open Mobile Environments and their Business Applications

Nor Shahniza Kamal Bashah [1] Natalia Kryvinska[2] and Do van Thanh[3]

[1] Norwegian University of Science and Technology, Department of Telematics, O.S. Bragstads Plass 2E, NO-7491 Trondheim, Norway – nor@item.ntnu.no
[2] Department of e-Business, Faculty of Business, Economics and Statistics, University of Vienna, Bruenner Str. 72, A-1210 Wien – natalia.kryvinska@univie.ac.at
[3] Telenor & NTNU, Snaroyveien 30, 1331 Fornebu, Norway – thanh-van.do@telenor.com

Abstract. In open mobile environments, mobile device may be connected to several network system offering heterogeneous sets of services using different names. The services can be anything and introduced by anybody, which results to have the possibility of having similar service with different names or different services having the same name. The current IEEE 802.21 Media Independent Handover Services facilitate handover across these heterogeneous networks. However, the standard just provides network continuity without ensuring service continuity during the handover. We propose extensions to the MIH, which enable the exchange of service information in addition to network information at handover. This service information constitutes the fundament for a novel service discovery, which is capable of discovering equivalent services and enabling the handover between them. In this paper, the extended MIH handover procedure is explained thoroughly. The additional service information elements are also described. The proposed service discovery system developed on Android is presented and a few use scenarios are given for illustration.

Keywords: service continuity, service handover, service discovery, service continuity business model, service discovery business application.

1. Introduction

In an age of convergence where media, information, service and network technologies are converging mobile phones are no longer terminals strictly tied to mobile telecommunication networks but have also the possibility to connect to data networks like Bluetooth, WLAN, WiMAX, etc. These data networks can be home, enterprise or public networks like airport, train station, shopping center, gas station, etc. The services offered on these networks, although originally different from the mobile networks, are now converging. Indeed, even telephony, a typical telecommunication service, is now offered on data networks using IP technologies. Although perceived as equivalent or comparable to the original telephony by the user, it is still technically considered as a different service and service continuity based on the combination of these two telephony services is consequently neglected.

The continuity of the telephony service between the telecommunication and data networks is quite desirable when the mobile phone is moving and the availability of networks and services is changing. To ensure smooth handover between heterogeneous networks, IEEE has specified the 802.21 Media Independent Handover (MIH) Services [1], which provide necessary information about the access networks such as availability, QoS resources, etc. Unfortunately, MIH is only aiming at preserving connections, which is insufficient to ensure service continuity for telephony realized by different technologies. The services are not considered as equivalent and there is no way to replace one service with an equivalent one. Typically, a mobile phone with an ongoing telephone conversation using IP telephony on a WLAN domain will lose this conversation when moving out of the WLAN domain and going into to the mobile network. To ensure service continuity, service information has to be provided to the mobile device in addition to the network information as proposed by the MIH.

In this paper, in order to ensure seamless service continuity across multiple heterogeneous networks the IEEE 802.21 MIH is proposed to be extended with service information and a sound service discovery capable of finding equivalent services in acceptable amount of time. The paper starts with a brief description of the IEEE 802.21 MIH. Next is the explanation of limitations of the MIH concerning service continuity. The main part of the paper explains the proposed extensions to the MIH handover procedure and to the Media Independent Information Service (MIIS). The mobile service discovery system developed on Android is presented and few use scenarios are given to illustrate the effectiveness of the proposed service discovery and matching procedure. Related works are then discussed and finally the paper is concluded with a summary of our research work and suggestions for further works.

2. Overview of IEEE 802.21 Media Independent Handover (MIH) Services

The IEEE standard 802.21 Media Independent Handover Services defines media access independent mechanisms that enable the optimization of handover between heterogeneous 802 systems and facilitate handover between 802 systems and cellular systems.

Fig. 1 shows an example of a network offering MIH services. A multi-access Mobile Node equipped with MIH functionality can connect to multiple Access network i.e. WiMAX, 2G/3G or wired Ethernet 802.3 via multiple Point of Access (PoA) and have access to MIH services at a Point of Service (PoS) provided by the Home MIH Information Server. This server includes an Information Database containing information about networks in the vicinity of the Mobile Network. When travelling the Mobile Node can communicate the visited MIH Information Server to acquire network information necessary for carrying out handover and preserving connectivity.

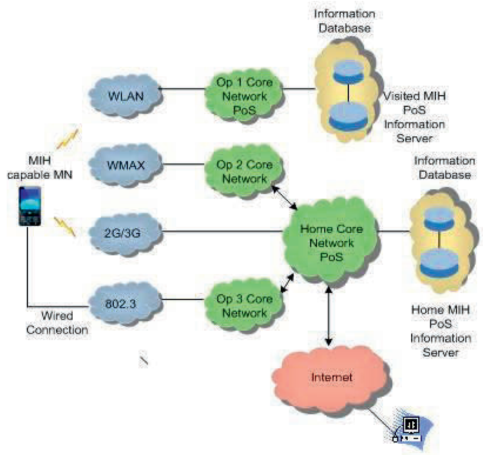

Fig.1. Example of network model with MIH services.

Fig. 2 shows the IEEE 802.21 MIH Function on layer 2 which provides three main services to layer 3 and upper as follows:

- **Media Independent Event Services (MIES)** which detects and notifies changes in link-layer properties.
- **Media Independent Command Service (MICS)** which provides a set of commands for the MIH users to control link properties.
- **Media Independent Information Service (MIIS)** which provides the information about different networks and their services thus enabling more effective handover decision to be made across heterogeneous networks.

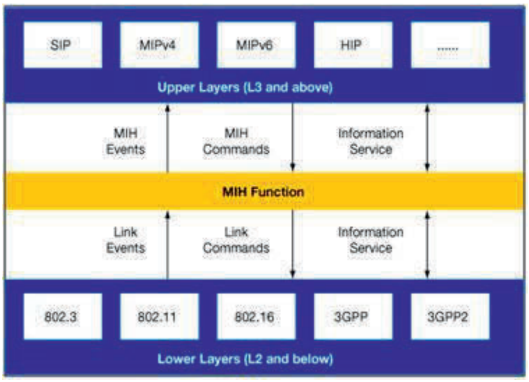

Fig.2. The IEEE 802.21 MIH architecture.

The MIH Function (MIHF) provides services to the layer 3 and upper through a single media independent interface (MIH SAP) and obtains services from the lower layers through a variety of media independent interfaces (media specific SAPs). Instead of having to keep powering up each of its individual radios and establishing network connections for retrieving heterogeneous network information, which causes battery drain, the Mobile Node can have access to the services of the MIHF via well-defined Point of Services (PoS).

3. Problem for Service Continuity and Service Availability

Preserving connectivity is a necessary but not sufficient condition to ensure service continuity. Indeed, it is sufficient only when the connection links are changed but it is insufficient when both the connections and the components realising the service are changing due to the mobility of the Mobile Node. To elucidate this let us consider two service usage cases as follows:

- **Asynchronous services:** The user is quite used to a service offered by his home operator, called Taxi, which connects him directly to the appropriate taxi dispatch service depending on his location. For example, when being in Oslo the Oslo taxi dispatch service will be connected but the Trondheim dispatch will be used when he is in Trondheim. When being on the move, he runs onto area only covered by WLAN where an equivalent service called "drosje" is available. His favourite Taxi service will be unavailable because no Taxi service is found.
- **Synchronous services:** The user is having a conversation using GSM telephony and arriving to an area where there is no GSM coverage but only WiMAX coverage. With the MIH support the mobile phone can smoothly switch to WiMAX. Unfortunately, the WiMAX network does not support GSM telephony but offers only an IP telephony service called WiPhone. The Mobile Node does not have any information about the WiPhone and is hence not able to use it. The user's conversation is consequently broken down.

From the two described cases, it is evident that in addition to network information it is necessary to have service information to ensure service continuity.

4. Proposed Extensions to the MIH Handover Procedure

In order to implement the extension necessary for the support of service continuity in the case of handover between two different but equivalent service let us consider the Mobile-initiated handover procedure and identify the steps where addition or modifications are to be done.

The Mobile-initiated Handover Procedure operates as follows:

- Step 1: Mobile Node is connected to the serving network via Current Point of Service (PoS) and it has access to MIH Information Server.

- Step 2: Mobile Node queries information about neighboring networks by sending the MIH_Get_Information request to Information Server. Information Server responds with MIH_Get_Information response. This information query may be attempted as soon as Mobile Node is first attached to the network.
- Step 3: Mobile Node triggers a mobile-initiated handover by sending MIH_MN_HO_Candidate_Query request to Serving PoS. This request queries information of potential candidate networks. It includes queries on QoS resources and/or IP address configuration method supported in the candidate networks.
- Step 4: Serving PoS queries the availability of resources at the candidate networks by sending MIH_N2N_HO_Query Resources request to one or multiple Candidate PoSs.
- Step 5: Candidate PoSs respond with MIH_N2N_HO_Query_Resources response and Serving PoS notifies the Mobile Node of the resulting resource availability at the candidate networks through MIH_MN_HO_Candidate_Query response.
- Step 6: Mobile Node decides the target of the handover and requests resource preparation by sending the MIH_MN_HO_Commit request to Serving PoS.
- Step 7: Serving PoS sends MIH_N2N_HO_Commit request to Target PoS to request resource preparation at the target network. Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit response.
- Step 8: When the resource is successfully prepared, Serving PoS sends MIH_MN_HO_Commit response to Mobile Node.
- Step 9: New layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network.
- Step 10: Mobile Node may send MIH_MN_HO_Complete request to Target PoS. Target PoS sends MIH_MN_HO_Complete request to previous Serving PoS to release resource which was allocated to Mobile Node. After identifying that resource is successfully released, Target PoS may send MIH_MN_HO_Complete response to Mobile Node.

To support service continuity the following extensions are proposed:

Step 3: Extension is needed in step 3if the Mobile Node is looking for a particular set of services and the candidate networks must support them.

The primitive MIH_MN_HO_Candidate_Query request has to be modified to accommodate service information, which constitutes one of the criteria for identifying candidate networks. We propose to introduce an additional parameter called QueryServiceList as follows:

```
MIH_MN_HO_Candidate_Query.request (
                            DestinationIdentifier,
                            CurrentLinkIdentifier,
                            CandidateLinkList,
                            QueryResourceList,
                            IPConfigurationMethods,
                            DHCPServerAddress,
                            FAAddress,
                            AccessRouterAddress,
                            QueryServiceList,
                            )
```

The parameter QueryServiceList is the list of services that the Mobile Node is requesting. It has a type LIST(SERVICE) which contains 0 or more services. The data type SERVICE is defined as shown in Table 1.

Table 1. Definition of Data Type Service.

Data type name	Derived from	Definition
SERVICE	SEQUENCE(	A type to represent a list of service in the

	SERVICE_NAME, SERVICE_TYPE, SERVICE_PARENT _TYPE, SERVICE_EQUIVA LENCE_CLASS, SERVICE_KEWOR DS, SERVICE_DESCRIP TION)	access network.
SERVICE_NAME	OCTET_STRING	A type to represent a service name. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_TYPE	OCTET_STRING	A type to represent in which category the service belongs to. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_PAREN T_TYPE	OCTET_STRING	A type to represent ParentType of a service. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_EQUIV ALENCE_CLASS	OCTET_STRING	A type to represent equivalence services (any EquivalenceClass of the service defined by the Service Provider; it can also be service with the same ParentType). The value is a non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_KEYW ORDS	OCTET_STRING	A type to represent keywords of a service. The value is a non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_DESCRI PTION	OCTET_STRING	A type to represent the description of a service. A non-NULL terminated string whose length shall not exceed 253 octets.

SERVICE_NAME identifies a service instance. A service instance has a unique SERVICE_NAME but a SERVICE_NAME may be given to several service instances. In a future mobile environment where anybody can be service provider and a service can be anything a SERVICE_NAME will not be standardized and the same name can be used by multiple service providers to denote multiple different services [2].

SERVICE_TYPE identifies the type of the service. As for SERVICE_NAME the SERVICE_TYPE can be ambiguous in a future mobile environment.

SERVICE_PARENT_TYPE identifies parent service type that the current service type is derived from. The SERVICE_PARENT_TYPE is necessary in the service matching when a partial match, i.e. only a subset of features is required [3].

SERVICE_EQUIVALENCE_CLASS identifies the equivalence class that the service type belongs to. This identifier is necessary to enable the usage of multiple languages to denote a service.

SERVICE_KEYWORDS contains attributes that helps narrow the scope of the service matching.

The SERVICE_DESCRIPTION contains information needed to use the service.

Step 4: It is necessary to modify the primitive MIH_N2N_HO_Query_Resources request to accommodate service information. We propose to introduce an additional parameter called ServiceRequirements as follows:

```
MIH_N2N_HO_Candidate_Query.request (
                    DestinationIdentifier,
                    QoSResourceRequirements,
                    IPConfigurationMethods,
```

DHCPServerAddress,
FAAddress,
AccessRouterAddress,
CandidateLinkList,
ServiceRequirements,
)

The parameter ServiceRequirements is the service requirements that the Mobile Node is requesting. It has a type LIST(SERVICE) which contains 0 or more services.

Step 5: In this step the list of candidate networks will be returned by Candidate PoS with the primitive MIH_N2N_HO_Query_Resources response and forwarded to the Mobile Node through the primitive MIH_MN_HO_Candidate_Query response. No modification is required for these two primitives but in order to compile the candidate list the Candidate PoSs will have to carry out a service discovery and matching in addition to the standard resource query.

Step 6: To select the target network for handover, the Mobile Node must choose the most appropriate available services and requests resource preparation by sending the MIH_MN_HO_Commit request to Serving PoS. In addition to resource preparation appropriate actions must be carried out on the network side to perform the service transfer to replace current service instance with new equivalent service instance.
The primitive MIH_MN_HO_Commit request must be extended with an additional parameter TargetServiceInfo as follows:

MIH_MN_HO_Commit.request (
DestinationIdentifier,
LinkType,
TargetNetworkInfo,
TargetServiceInfo,
)

The parameter TargetServiceInfo has LIST(SERVICE) as type and contains the names of services to be prepared for the handover.

Step 7: Serving PoS sends MIH_N2N_HO_Commit request to Target PoS to request resource preparation at the target network.
The primitive has to be extended with an additional parameter RequestedServiceSet of LIST(SERVICE) type which indicates the services that needs to be prepared for handover as follows:

MIH_N2N_HO_Commit.request (
DestinationIdentifier,
MNIdentifier,
TargetMNLinkIdentifier,
TargetPoA,
RequestedResourceSet,
RequestedServiceSet,
)

Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit response, which is extended with the parameter AssignedServiceSet of LIST(SERVICE) type as follows:

MIH_N2N_HO_Commit.response (
DestinationIdentifier,
Status,
MNIdentifier,
TargetLinkIdentifier,
AssignedResourceSet,
AssignedServiceSet,
)

Step 8: When the resource is successfully prepared, Serving PoS sends MIH_MN_HO_Commit response to Mobile Node. The primitive MIH_MN_HO_Commit.response is also extended with an additional parameter called TargetServiceInfo of LIST(SERVICE) type as follows:

MIH_MN_HO_Commit.response (
DestinationIdentifier,
Status,
LinkType,
TargetServiceInfo,
)

Step 9: New layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network. In addition to these standard procedures for the MIH, we propose to perform also service transfer procedures to ensure service continuity.

Step 10: Mobile Node may send MIH_MN_HO_Complete request to Target PoS. Target PoS sends MIH_N2N_HO_Complete request to previous Serving PoS to release resource which was allocated to Mobile Node. After identifying that resource is successfully relased, Target PoS may send MIH_MN_HO_Complete response to Mobile Node. In addition to the release of radio resources it may also be necessary to release service resources.

To enable service continuity the Media Independent Command Service (MICS) must accommodate the modified commands as described above.

5. Proposed Extension to the Media Independent Information Service (MIIS)

The Media Independent Information Service (MIIS) provides a framework and corresponding mechanisms by which an MIHF entity can discover and obtain network information existing within a geographical area to facilitate handovers. To ensure service continuity it is also necessary with extensions in the MIIS as proposed in [4]. More specifically, new Information Elements (IEs) containing service information must be added to the standard set as follows:

- IE_NET_SERVICES introduced in the Access Network Container which contains information about services supported by the access network. This element is used when the network is offering the same services at every PoAs. If the network does provide different services at different PoAs, this information element must be left empty. When encountering an empty IE_NET_SERVICES the service discovery will have to proceed to the respective PoA container to carry out the service matching.
- IE_POA_SERVICES introduced in the PoA Container which contains information on services supported at the PoA. For networks offering the same services at all the PoAs, this element must be empty since the information about the services is already specified at the network level.

Fig. 3 depicts the example on how the new service information is included in the Information Elements.

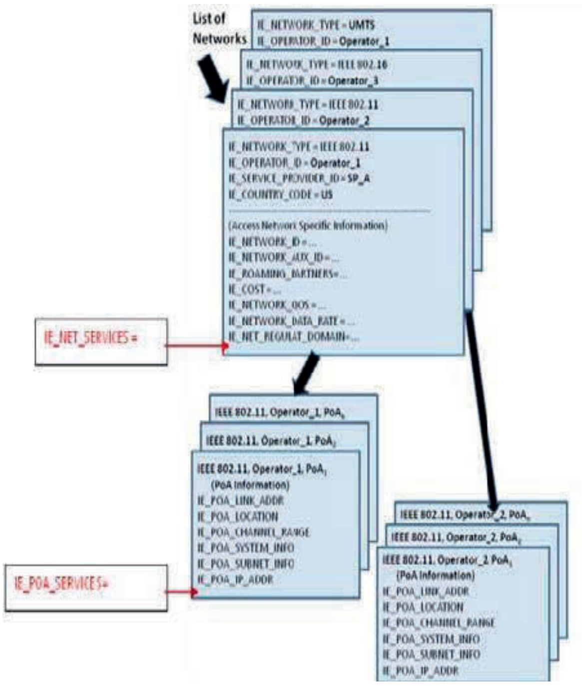

Fig. 3. Information elements with service information.

6. Implementation of Service Discovery

In this section the service discovery development environment is described. The user interface is explained and four use scenarios are given to illustrate the effectiveness of the proposed service discovery and matching procedure.

6.1. Mobile Service Discovery Testbed

The service discovery application is developed in the Java™ [5] language using the Android Software Development Kit (SDK) which provides the tools and APIs necessary to begin developing applications on the Android platform. Android [6] is a software stack for mobile devices that includes an operating system, middleware and key applications. Android platform is chosen because it provides access to a wide range of useful libraries and tools that can be used to build rich applications.

Eclipse Helios [7] is used for the IDE and Android plugin for Eclipse is also needed for the system development. The developed application is flexible and capable of handling all the request variants from the Mobile Node. Android emulator is used for the testing purpose. The application can be installed and run in any types of mobile phones with Android 2.3 operating system and later.

6.2. System User Interface

Error! Reference source not found. 4 shows the icon of the service discovery in the Android emulator. **Error! Reference source not found.** 5 depicts the service discovery user interface offering three options to find service as follows:

1) by Service Name - the EquivalenceClass semantic will be used and all the similar services with different names and languages will be return.
2) by Service Type - all the subtype services under the same ParentType will be return.

3) by Keyword - even the non-exact match services will be return as long as it has something in common to describe about the service (for e.g. Snacks or Cocktail to define Food service).

Fig. 4. Service discovery icon on Android emulator.

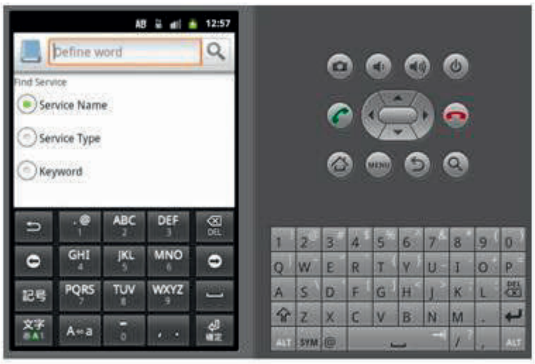

Fig. 5. Service discovery user interface.

6.3. Use scenarios

Use scenario 1 - Service discovery and matching based on the service name.

The Mobile Node may request for a specific service by specifying its service name. For example, if the Mobile Node requests for a Skype service, only the networks offering Skype service should be returned. It is worth noting that a network with a completely different service with the name of Skype can be returned because the service names are not standardized and unique. To avoid this situation the Mobile Node may have to specify the service type in addition to the service name to aid the service discovery. For example, when the Mobile Node requests a service with the service name Book and specify Online Business as service type as illustrated in **Error! Reference source not found.**, only the Book service with the service type Online Business is returned.

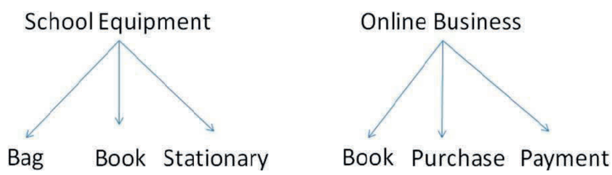

Fig.6. Example of Book service denotes different type of services.

If the Mobile Node specifies only the service type without the service name all the supported services with the same service type will be return. For example, if the Mobile Node requests for IP telephony service, Skype, VoIP and WiPhone, which has IP Telephony as same service type will be returned.

Use scenario 2 - Service discovery and matching based on ParentType.

The Mobile Node may also request a service by specifying the ParentType. All the services with the same ParentType will be returned. In this case only partial match is required. For example, if the Mobile Node specifies Telephony as ParentType all the child service types of Telephony e.g. Skype, GSM, VoIP and WiPhone will be returned as illustrated in **Error! Reference source not found.**

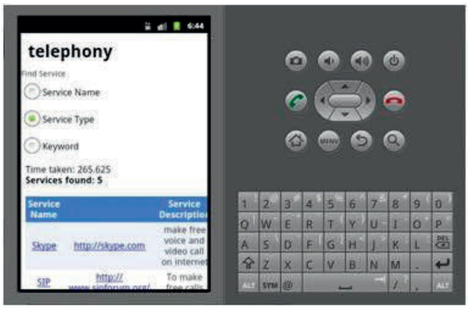

Fig. 7. Service discovery and matching based on ParentType.

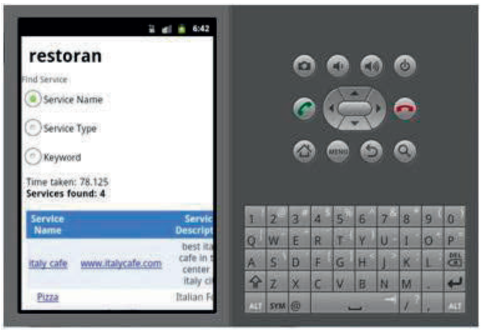

Fig. 8. Service discovery and matching based on EquivalenceClass.

Use scenario 3 - Service discovery and matching based on EquivalenceClass.

The Mobile Node may request a service by specifying the EquivalenceClass. In this case all the services of equivalent service type will be returned. The returned services can have different names, types in different languages. For example, if the Mobile Node requests service by specifying Restoran as EquivalenceClass, Café, Bistro, Restaurant, etc. will be returned as illustrated in Fig. 8.

Use scenario 4 - Service discovery and matching based on Keywords.

The Mobile Node can request a service by specifying Keywords. This can be used especially when the user has only a vague idea about the service he wants to request. The service matching the Keywords will be returned. For example, if the Mobile Node wants to find a Food service, it may send Meal, Snacks or Dining as Keywords to request for the Food service as illustrated in Fig. 9.

Fig. 9. Service discovery and matching based on Keywords.

The time taken in the output screen indicates the processing time (in seconds) to retrieve the search result.

7. Related Work

Semantic service discovery has been widely discussed and addressed by several research initiatives that have proposed enhanced discovery mechanisms. An Integrated Semantic Service Discovery proposed by Shanshan [8] enhances the discovery ability by making use of service ontology. It takes into account both the functional and non-functional properties to achieve accurate and satisfactory discovery results. Ontologies are defined in the Web Ontology Language, OWL [9]. Behavioral semantics are added to WSDL file by associating service functionality related elements with links to OWL-based service ontology. Non-functional properties are specified as QoS parameters and rule-based policies comprising business policies, QoS policies and context policies.

Efficient Routing Grounded on Taxonomy (ERGOT) [10] is a semantic-based system for service discovery in distributed infrastructures. It is a system that combines Distributed Hash Tables (DHTs) and Semantic Overlay Networks (SONs) to enable semantic-based service discovery in distributed infrastructures such as Grids and Clouds. ERGOT takes advantage of semantic annotations that enrich service specifications in two ways: (i) services are advertised in the DHT on the basis of their annotations, thus allowing establishing a SON among service providers; (ii) annotations enable semantic-based service matchmaking, using a novel similarity measure between service requests and descriptions.

UbiSearch [11] is a semantic service discovery network for large-scale ubiquitous computing environments. A semantic service discovery network in the semantic vector space is proposed where services that are semantically close to each other are mapped to nearby positions so that the similar services are registered in a cluster of resolvers. Using this mapping technique, the search space for a query is efficiently confined within a minimized cluster region while maintaining high accuracy in comparison to the centralized scheme. It supports scalable semantic queries with low communication overhead, balanced load distribution among resolvers for service registration and query processing and personalised semantic matching.

An Ontology-enhanced cloud service discovery system (CSDS) [12] aims to support the Cloud users in finding a Cloud service over the Internet. The CSDS interacts with Cloud ontology to determine the similarities between and among services. The Cloud Service Reasoning Agent (CSRA) determines the relations of Cloud services using three service reasoning methods which are similarity reasoning, equivalent reasoning and numerical reasoning.

In [13] a hierarchical and chord-based semantic service discovery system is proposed for the universal network (UniNServ). It uses OWL-S (Web Ontology Language for Services) to describe services and adopts Chord [14] as a distributed lookup protocol. Besides, UniNServ uses three types of ontologies to perform automatic semantic service discovery with QoS through exploiting the logical relationships within the services. They append QoS measurements to OWL-S and called it OWL-QoS.

MEMORY [15] is a matrix-based efficient semantic web service (SWS) discovery system which does ontological pre-reasoning and holds the reasoning results in matrix forms in service publishing phase, so that it can transfer the load of semantic reasoning from service query to service publication and perform fast matching during service discovery. A novel OWL-S (formerly DAML-S) based semantic service discovery system is proposed for dynamically discovering complex constraint-based services.

The main objective of the above mentioned semantic service discovery systems are to deliver and find the right service to the user in an acceptable time. Thus it has become an important factor for an efficient service discovery system but we propose a more comprehensive and novel service discovery system in open mobile environments which is capable of finding services having arbitrary names in any format given by any service provider.

8. Conclusion

In this paper, it is explained when and how the IEEE 802.21 MIH fails to support service continuity. To remedy the situation, it is proposed to exchange information about supported services in addition to the network QoS information. The paper also indicates the necessary extensions in the handover procedure and specifies the additional information elements containing service information. A novel service discovery has also been proposed as additional function for the MIH Information Servers. To verify the feasibility of the proposed extension, a simplified and simulated MIH containing only service information and service capability has been successfully implemented [16]. The MIH capability including service continuity support on the Mobile Node has also successfully developed on Android and illustrated by few use scenarios in this paper. For further work the proposed extension in the existing IEEE 802.21 MIH Services should be implemented and the service discovery should be installed in any MIH client so that testing and evaluation can be carried out in the real mobile environment.

References

1. IEEE Computer Society, "IEEE Standard for Local and Metworpolitan Area Networks. Part 21: Media Independent Handover Service", IEEE Std. 802.21™ – 2008. 21 January 2009.
2. Kamal Bashah, N.S, Jørstad, I. & van Do, T, "Service Discovery in Future Open Mobile Environments", Proc. of the 4th International Conference on Digital Society (ICDS2010), IEEE Conference Proceedings, February 2010, ISBN 978-0-7695-3953-9.
3. Kamal Bashah, N.S, Kryvinska, N. & van Do, T, "Service Discovery in Ubiquitous Mobile Computing Environment", Proc. of the 12th International Conference on Information Integration and Web-based Applications & Services (IIWAS2010), ACM Publication, November 2010, ISBN 978-1-4503-0440-5.
4. Kamal Bashah, N.S, Jørstad, I. & van Do, T, "Enabling Service Continuity on Future Mobile Services", Proc. of the International Symposium on Wireless and Pervasive Computing (ISWPC2009), IEEE Conference Proceedings, February 2009, ISBN 978-1-4244-2965-3.
5. G. James, J. Bill and S. Guy, *The Java Language Specification;* Third Edition: Addison Wesley, 2005.
6. Android developers – http://developer.android.com/index.html.
7. Eclipse - http://www.eclipse.org.
8. S. Jiang and F. Aagesen, "An Approach to Integrated Semantic Service Discovery Autonomic Networking." vol. 4195, D. Gaïti*, et al.*, Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 159-171.
9. O. W. Group, "OWL Web Ontology Language Overview ", ed, 2004.
10. Pirr*, et al.*, "ERGOT: A Semantic-Based System for Service Discovery in Distributed Infrastructures," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, 2010, pp. 263-272.
11. D. K. Saehoon Kang, Younghee Lee, Soon J. Hyun, Dongman Lee, and Ben Lee, "A Semantic Service Discovery Network for Large-Scale Ubiquitous Computing Environments," *ETRI Journal,* vol. 29, pp. pp.545-558, Oct. 2007 2007.
12. T. H. a. K. M. Sim, "An Ontology-enhanced cloud service discovery system," presented at the International MultiConference of Engineers and Computer Scientists 2010 (IMECS2010), Hong Kong, 2010.
13. H. H. Ying Zhang, Dong Yang and Hongke Zhang, "A hierarchical and chord-based semantic service discovery system in the universal network," *International Journal of Innovative Computing, Information and Control,* vol. 5, November 2009 2009.

14. I. Stoica*, et al.*, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *Networking, IEEE/ACM Transactions on,* vol. 11, pp. 17-32, 2003.
15. Z. Zhao*, et al.*, "MEMORY: A Matrix-Based Efficient Semantic Web Service Discovery System," presented at the Proceedings of the 2009 Fifth International Conference on Next Generation Web Services Practices, 2009.
16. Kamal Bashah, N.S, Bhatti, A, Choudhary, I.A, Jørstad, I. & van Do, T, "Service Discovery for Mobile Multi-domain Multi-language Environments", Proc. of the 3rd IEEE International Workshop on Selected Topics in Mobile and Wireless Computing (STMWC2010), IEEE Computer Society, October 2010, ISBN ISBN 978-1-4244-7741-8.

PAPER 10: Quality-driven Service Discovery Techniques for Open Mobile Environments and their Business Applications

Nor Shahniza Kamal Bashah, Natalia Kryvinska & Do van Thanh

Journal of Service Science Research, Journal No. 12927 *(accepted)*

ISSN 2093-0720 (print version)

ISSN 2093-0739 (electronic version)

Quality-driven Service Discovery Techniques for Open Mobile Environments and their Business Applications

Nor Shahniza Kamal Bashah, Natalia Kryvinska, and Do van Thanh

ABSTRACT

In open mobile environments, mobile device may be connected to several network system offering heterogeneous sets of services using different names. The services can be anything and introduced by anybody, which results to have the possibility of having similar service with different names or different services having the same name. The current IEEE 802.21 Media Independent Handover Services facilitate handover across these heterogeneous networks. However, the standard just provides network continuity without ensuring service continuity during the handover. We propose extensions to the MIH, which enable the exchange of service information in addition to network information at handover. This service information constitutes the fundament for a quality-driven service discovery, which is capable of discovering equivalent services and enabling the handover between them. In this paper, the extended MIH handover procedure is explained thoroughly. The additional service information elements are also described. The proposed service discovery system developed on Android is presented and a few use scenarios are given for illustration.

KEYWORDS

Service continuity, Service handover, Service discovery, Service continuity business model, Service discovery business application.

1. INTRODUCTION

In an age of convergence where media, information, service and network technologies are converging mobile phones are no longer terminals strictly tied to mobile telecommunication networks but have also the possibility to connect to data networks like Bluetooth, WLAN, WiMAX, etc. These data networks can be home, enterprise or public networks like airport, train station, shopping center, gas station, etc. The services offered on these networks, although originally different from the mobile networks, are now converging. Indeed, even telephony, a typical telecommunication service, is now offered on data networks using IP technologies. Although perceived as equivalent or comparable to the original telephony by the user, it is still technically considered as a different service and service continuity based on the combination of these two telephony services is consequently neglected.

The continuity of the telephony service between the telecommunication and data networks is quite desirable when the mobile phone is moving and the availability of networks and services is changing. To ensure smooth handover between heterogeneous networks, IEEE has specified the 802.21 Media Independent Handover (MIH) Services (IEEE Computer Society 2009), which provide necessary information about the access networks such as availability, QoS resources, etc. Unfortunately, MIH is only aiming at preserving connections, which is insufficient to ensure service continuity for telephony realized by different technologies. The services are not considered as equivalent and there is no way to replace one service with an equivalent one. Typically, a mobile phone with an ongoing telephone conversation using IP telephony on a WLAN domain will lose this conversation when moving out of the WLAN domain and going into to the mobile network. To ensure service continuity, service information has to be provided to the mobile device in addition to the network information as proposed by the MIH.

In this paper, in order to ensure seamless service continuity across multiple heterogeneous networks the IEEE 802.21 MIH is proposed to be extended with service information and a sound service discovery capable of finding equivalent services in acceptable amount of time. The paper starts with some explanation on the related works followed by a brief description of the IEEE 802.21 MIH. Next is the explanation of limitations of the MIH concerning service continuity. The main part of the paper explains the proposed extensions to the MIH handover procedure and to the Media Independent Information Service (MIIS). The mobile service discovery system developed on Android is presented and few use scenarios are given to illustrate the effectiveness of the proposed service discovery and matching procedure. The paper is concluded with a summary of our research work and suggestions for further works.

2. RELATED WORK

Semantic service discovery has been widely discussed and addressed by several research initiatives that have proposed enhanced discovery mechanisms. An Integrated Semantic Service Discovery proposed by Shanshan (S. Jiang and F. Aagesen 2006) enhances the discovery ability by making use of service ontology. It takes into account both the functional and non-functional properties to achieve accurate and satisfactory discovery results. Ontologies are defined in the Web Ontology Language, OWL (OWL Group 2004). Behavioral semantics are added to WSDL file by associating service functionality related elements with links to OWL-based service ontology. Non-functional properties are specified as QoS parameters and rule-based policies comprising business policies, QoS policies and context policies.

Efficient Routing Grounded on Taxonomy (ERGOT) (Pirr, et al 2010) is a semantic-based system for service discovery in distributed infrastructures. It is a system that combines Distributed Hash Tables (DHTs) (B Wiley 2003) and Semantic Overlay Networks (SONs) (K. Aberer, et al 2005) to enable semantic-based service discovery in distributed infrastructures such as Grids and Clouds. ERGOT takes advantage of semantic annotations that enrich service specifications in two ways: (i) services are advertised in the DHT on the basis of their annotations, thus allowing establishing a SON among service providers; (ii) annotations enable semantic-based service matchmaking, using a novel similarity measure between service requests and descriptions.

UbiSearch (D. K. Saehoon Kang, et al 2007) is a semantic service discovery network for large-scale ubiquitous computing environments. A semantic service discovery network in the semantic vector space is proposed where services that are semantically close to each other are mapped to nearby positions so that the similar services are registered in a cluster of resolvers. Using this mapping technique, the search space for a query is efficiently confined within a minimized cluster region while maintaining high accuracy in comparison to the centralized scheme. It supports scalable semantic queries with low communication overhead, balanced load distribution among resolvers for service registration and query processing and personalised semantic matching.

An Ontology-enhanced cloud service discovery system (CSDS) (T. H. a. K. M. Sim 2010) aims to support the Cloud users in finding a Cloud service over the Internet. The CSDS interacts with Cloud ontology to determine the similarities between and among services. The Cloud Service Reasoning Agent (CSRA) determines the relations of Cloud services using three service reasoning methods which are similarity reasoning, equivalent reasoning and numerical reasoning.

A hierarchical and chord-based semantic service discovery system is proposed for the universal network (UniNServ) (H. H. Ying Zhang, et al 2010). It uses OWL-S (Web Ontology Language for Services) to describe services and adopts Chord (I. Stoica, et al 2003) as a distributed lookup protocol. Besides, UniNServ uses three types of ontologies to perform automatic semantic service discovery with QoS through exploiting the logical relationships within the services. They append QoS measurements to OWL-S and called it OWL-QoS.

MEMORY (Z. Zhao, et al 2009) is a matrix-based efficient semantic web service (SWS) discovery system which does ontological pre-reasoning and holds the reasoning results in matrix forms in service publishing phase, so that it can transfer the load of semantic reasoning from service query to service publication and perform fast matching during service discovery. A novel OWL-S (formerly DAML-S) based semantic service discovery system is proposed for dynamically discovering complex constraint-based services.

A novel OWL-S (formerly DAML-S) based semantic service discovery system is proposed by Le Duy Ngan et. al (L. D. Ngan, et al 2011) for dynamically discovering complex constraint-based services. It is based on representing complex service constraints as Semantic Web Rule Language (SWRL) rules and using a rule engine for matchmaking and handling dynamism via a real-time ontology population and reasoning infrastructure. It supports descriptions with complex constraints and matching of service providers and requesters dynamically.

A layered architecture of semantic service discovery system is proposed by incorporating search crawler as a core component for discovering services resided on the provider websites (V. Kaewmarin, et al 2008). The search crawler can operate in multi-threaded environment to enhance the capability of discovering a number of distributed Web services simultaneously, as well as in various UDDI registries. In order to enrich service discovery in a semantic manner, the Web services descriptions returned from the search crawlers are thus modeled into a machine-processable representation language such as OWL-S. The system conformed to the proposed architecture will provide the flexibility and extensibility to accomplish complex Web service requests that meet user-specified functional requirements.

A two stages method of semantic service discovery is proposed (P. Yan-Bin, et al 2009) whereby the service is divided into two classes - core service description registered at core service library (CSL) and assistant service which is registered at assistant service library (ASL). The service requirement is send to two stages service discovery module by service requester and the module discover target service through two stages: 1) in core service discovery stage - find target core service according to similarity of service description between service requirement and core service; 2) in assistant service composition stage - form target composite service through composing target core service with related assistant services.

The main objective of the above mentioned semantic service discovery systems are to deliver and find the right service to the user in an acceptable time. Thus it has become an important factor for an efficient service discovery system but we propose a more comprehensive and novel service discovery system in open mobile environments which is capable of finding services having arbitrary names in any format given by any service provider.

3. OVERVIEW OF IEEE 802.21 MEDIA INDEPENDENT HANDOVER (MIH) SERVICES

The IEEE standard 802.21 Media Independent Handover Services defines media access independent mechanisms that enable the optimization of handover between heterogeneous 802 systems and facilitate handover between 802 systems and cellular systems.

Figure 1 shows an example of a network offering MIH services. A multi-access Mobile Node equipped with MIH functionality can connect to multiple Access network i.e. WiMAX, 2G/3G or wired Ethernet 802.3 via multiple Point of Access (PoA) and have access to MIH services at a Point of Service (PoS) provided by the Home MIH Information Server. This server includes an Information Database containing information about networks in the vicinity of the Mobile Network. When travelling the Mobile Node can communicate the visited MIH Information Server to acquire network information necessary for carrying out handover and preserving connectivity.

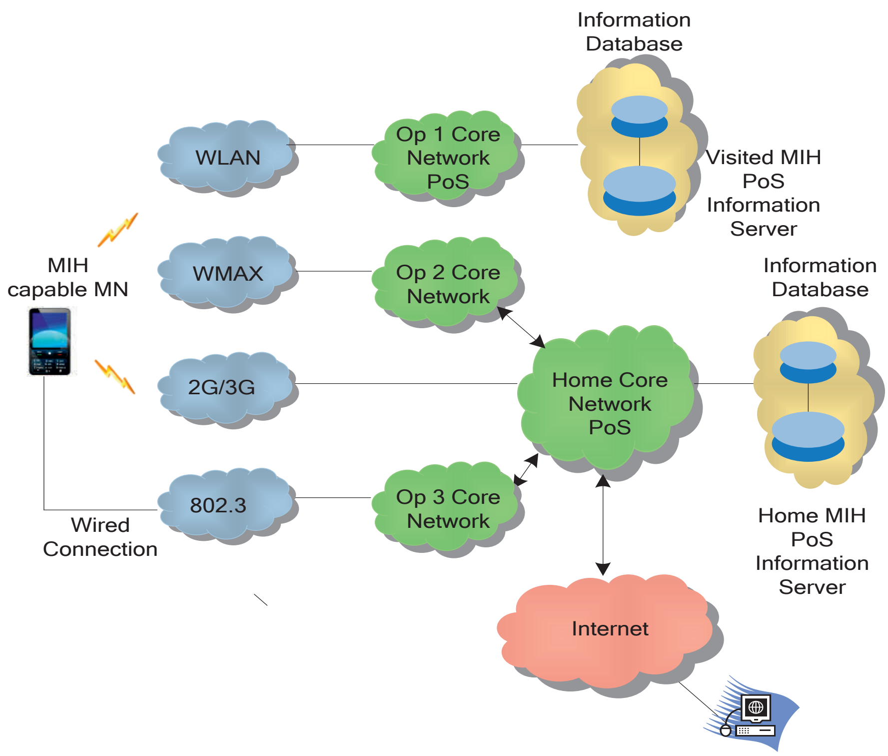

Figure 1. Example of network model with MIH services

Figure 2 shows the IEEE 802.21 MIH Function on layer 2 which provides three main services to layer 3 and upper as follows:

- **Media Independent Event Services (MIES)** which detects and notifies changes in link-layer properties.
- **Media Independent Command Service (MICS)** which provides a set of commands for the MIH users to control link properties.
- **Media Independent Information Service (MIIS)** which provides the information about different networks and their services thus enabling more effective handover decision to be made across heterogeneous networks.

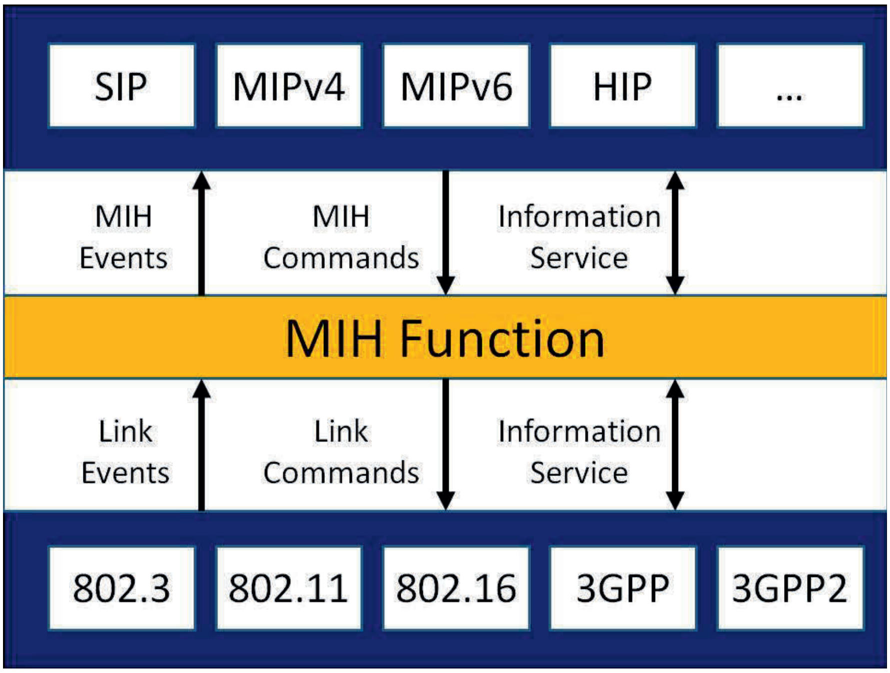

Figure 2. The IEEE 802.21 MIH architecture

The MIH Function (MIHF) provides services to the layer 3 and upper through a single media independent interface (MIH SAP) and obtains services from the lower layers through a variety of media independent interfaces (media specific SAPs). Instead of having to keep powering up each of its individual radios and establishing network connections for retrieving heterogeneous network information, which causes battery drain, the Mobile Node can have access to the services of the MIHF via well-defined Point of Services (PoS).

4. PROBLEM FOR SERVICE CONTINUITY AND SERVICE AVAILABILITY

Preserving connectivity is a necessary but not sufficient condition to ensure service continuity. Indeed, it is sufficient only when the connection links are changed but it is insufficient when both the connections and the components realising the service are changing due to the mobility of the Mobile Node.

Ensuring service availability is also one of the important factors when offering services to the end users especially when it involves heterogeneous networks (N. Kryvinska, C. Strauss, P. Zinterhof 2011; Khan, et al 2011).

To elucidate this let us consider two service usage cases as follows:

- **Asynchronous services:** The user is quite used to a service offered by his home operator, called Taxi, which connects him directly to the appropriate taxi dispatch service depending on his location. For example, when being in Oslo the Oslo taxi dispatch service will be connected but the Trondheim dispatch will be used when he is in Trondheim. When being on the move, he runs onto area only covered by WLAN where an equivalent service called “drosje” is available. His favourite Taxi service will be unavailable because no Taxi service is found.
- **Synchronous services:** The user is having a conversation using GSM telephony and arriving to an area where there is no GSM coverage but only WiMAX coverage. With the MIH support the mobile phone can smoothly switch to WiMAX. Unfortunately, the WiMAX network does not support GSM telephony but offers only an IP telephony service called WiPhone. The Mobile Node does not have any information about the WiPhone and is hence not able to use it. The user’s conversation is consequently broken down.

From the two described cases, it is evident that in addition to network information it is necessary to have service information to ensure service continuity.

5. PROPOSED EXTENSIONS TO THE MIH HANDOVER PROCEDURE

In order to implement the extension necessary for the support of service continuity in the case of handover between two different but equivalent service let us consider the Mobile-initiated handover procedure and identify the steps where addition or modifications are to be done.

The Mobile-initiated Handover Procedure operates as follows:

- Step 1: Mobile Node is connected to the serving network via Current Point of Service (PoS) and it has access to MIH Information Server.

- Step 2: Mobile Node queries information about neighboring networks by sending the MIH_Get_Information request to Information Server. Information Server responds with MIH_Get_Information response. This information query may be attempted as soon as Mobile Node is first attached to the network.
- Step 3: Mobile Node triggers a mobile-initiated handover by sending MIH_MN_HO_Candidate_Query request to Serving PoS. This request queries information of potential candidate networks. It includes queries on QoS resources and/or IP address configuration method supported in the candidate networks.
- Step 4: Serving PoS queries the availability of resources at the candidate networks by sending MIH_N2N_HO_Query Resources request to one or multiple Candidate PoSs.
- Step 5: Candidate PoSs respond with MIH_N2N_HO_Query_Resources response and Serving PoS notifies the Mobile Node of the resulting resource availability at the candidate networks through MIH_MN_HO_Candidate_Query response.
- Step 6: Mobile Node decides the target of the handover and requests resource preparation by sending the MIH_MN_HO_Commit request to Serving PoS.
- Step 7: Serving PoS sends MIH_N2N_HO_Commit request to Target PoS to request resource preparation at the target network. Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit response.
- Step 8: When the resource is successfully prepared, Serving PoS sends MIH_MN_HO_Commit response to Mobile Node.
- Step 9: New layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network.
- Step 10: Mobile Node may send MIH_MN_HO_Complete request to Target PoS. Target PoS sends MIH_MN_HO_Complete request to previous Serving PoS to release resource which was allocated to Mobile Node. After identifying that resource is successfully released, Target PoS may send MIH_MN_HO_Complete response to Mobile Node.

To support service continuity the following extensions are proposed:

Step 3: Extension is needed in step 3if the Mobile Node is looking for a particular set of services and the candidate networks must support them.

The primitive MIH_MN_HO_Candidate_Query request has to be modified to accommodate service information, which constitutes one of the criteria for identifying candidate networks. We propose to introduce an additional parameter called QueryServiceList as follows:

MIH_MN_HO_Candidate_Query.request (

DestinationIdentifier,

CurrentLinkIdentifier,

CandidateLinkList,

QueryResourceList,

IPConfigurationMethods,

DHCPServerAddress,

FAAddress,

AccessRouterAddress,

QueryServiceList,

)

The parameter QueryServiceList is the list of services that the Mobile Node is requesting. It has a type LIST(SERVICE) which contains 0 or more services. The data type SERVICE is defined as shown in TABLE 1.

Table 1. Definition of Data Type Service.

Data type name	Derived from	Definition
SERVICE	SEQUENCE(SERVICE_NAME, SERVICE_TYPE, SERVICE_PARENT_TYPE, SERVICE_EQUIVALENCE_	A type to represent a list of service in the access network.

	CLASS, SERVICE_KEWORDS, SERVICE_DESCRIPTION)	
SERVICE_NAME	OCTET_STRING	A type to represent a service name. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_TYPE	OCTET_STRING	A type to represent in which category the service belongs to. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_PAREN T_TYPE	OCTET_STRING	A type to represent ParentType of a service. A non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_EQUIVA LENCE_CLASS	OCTET_STRING	A type to represent equivalence services (any EquivalenceClass of the service defined by the Service Provider; it can also be service with the same ParentType). The value is a non-NULL terminated string whose length shall not exceed 253 octets.
SERVICE_KEYWO RDS	OCTET_STRING	A type to represent keywords of a service. The value is a non-NULL terminated string whose length shall not exceed 253 octets.

SERVICE_DESCRIPTION	OCTET_STRING	A type to represent the description of a service. A non-NULL terminated string whose length shall not exceed 253 octets.

SERVICE_NAME identifies a service instance. A service instance has a unique SERVICE_NAME but a SERVICE_NAME may be given to several service instances. In a future mobile environment where anybody can be service provider and a service can be anything a SERVICE_NAME will not be standardized and the same name can be used by multiple service providers to denote multiple different services (Bashah, et al 2010).

SERVICE_TYPE identifies the type of the service. As for SERVICE_NAME the SERVICE_TYPE can be ambiguous in a future mobile environment.

SERVICE_PARENT_TYPE identifies parent service type that the current service type is derived from. The SERVICE_PARENT_TYPE is necessary in the service matching when a partial match, i.e. only a subset of features is required (Bashah, et al 2010).

SERVICE_EQUIVALENCE_CLASS identifies the equivalence class that the service type belongs to. This identifier is necessary to enable the usage of multiple languages to denote a service.

SERVICE_KEYWORDS contains attributes that helps narrow the scope of the service matching.

The SERVICE_DESCRIPTION contains information needed to use the service.

Step 4: It is necessary to modify the primitive MIH_N2N_HO_Query_Resources request to accommodate service information. We propose to introduce an additional parameter called ServiceRequirements as follows:

MIH_N2N_HO_Candidate_Query.request (
DestinationIdentifier,
QoSResourceRequirements,
IPConfigurationMethods,

DHCPServerAddress,
FAAddress,
AccessRouterAddress,
CandidateLinkList,
ServiceRequirements,
)

The parameter ServiceRequirements is the service requirements that the Mobile Node is requesting. It has a type LIST(SERVICE) which contains 0 or more services.

Step 5: In this step the list of candidate networks will be returned by Candidate PoS with the primitive MIH_N2N_HO_Query_Resources response and forwarded to the Mobile Node through the primitive MIH_MN_HO_Candidate_Query response. No modification is required for these two primitives but in order to compile the candidate list the Candidate PoSs will have to carry out a service discovery and matching in addition to the standard resource query.

Step 6: To select the target network for handover, the Mobile Node must choose the most appropriate available services and requests resource preparation by sending the MIH_MN_HO_Commit request to Serving PoS. In addition to resource preparation appropriate actions must be carried out on the network side to perform the service transfer to replace current service instance with new equivalent service instance.

The primitive MIH_MN_HO_Commit request must be extended with an additional parameter TargetServiceInfo as follows:

MIH_MN_HO_Commit.request (
DestinationIdentifier,
LinkType,
TargetNetworkInfo,
TargetServiceInfo,
)

The parameter TargetServiceInfo has LIST(SERVICE) as type and contains the names of services to be prepared for the handover.

Step 7: Serving PoS sends MIH_N2N_HO_Commit request to Target PoS to request resource preparation at the target network.

The primitive has to be extended with an additional parameter RequestedServiceSet of LIST(SERVICE) type which indicates the services that needs to be prepared for handover as follows:

MIH_N2N_HO_Commit.request (
    DestinationIdentifier,
    MNIdentifier,
    TargetMNLinkIdentifier,
    TargetPoA,
    RequestedResourceSet,
    RequestedServiceSet,
)

Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit response, which is extended with the parameter AssignedServiceSet of LIST(SERVICE) type as follows:

MIH_N2N_HO_Commit.response (
    DestinationIdentifier,
    Status,
    MNIdentifier,
    TargetLinkIdentifier,
    AssignedResourceSet,
    AssignedServiceSet,
)

Step 8: When the resource is successfully prepared, Serving PoS sends MIH_MN_HO_Commit response to Mobile Node. The primitive

MIH_MN_HO_Commit.response is also extended with an additional parameter called TargetServiceInfo of LIST(SERVICE) type as follows:

MIH_MN_HO_Commit.response (
DestinationIdentifier,
Status,
LinkType,
TargetServiceInfo,
)

Step 9: New layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network. In addition to these standard procedures for the MIH, we propose to perform also service transfer procedures to ensure service continuity.

Step 10: Mobile Node may send MIH_MN_HO_Complete request to Target PoS. Target PoS sends MIH_N2N_HO_Complete request to previous Serving PoS to release resource which was allocated to Mobile Node. After identifying that resource is successfully relased, Target PoS may send MIH_MN_HO_Complete response to Mobile Node. In addition to the release of radio resources it may also be necessary to release service resources.

To enable service continuity the Media Independent Command Service (MICS) must accommodate the modified commands as described above.

6. PROPOSED EXTENSIONS TO THE MEDIA INDEPENDENT INFORMATION SERVICE (MIIS)

The Media Independent Information Service (MIIS) provides a framework and corresponding mechanisms by which an MIHF entity can discover and obtain network information existing within a geographical area to facilitate handovers. To ensure service continuity it is also necessary with extensions in the MIIS as proposed in (Bashah, et al 2009). More specifically, new Information Elements (IEs) containing service information must be added to the standard set as follows:

- IE_NET_SERVICES introduced in the Access Network Container which contains information about services supported by the access network. This element is used when the network is offering the same services at every PoAs. If the network does provide different services at different PoAs, this information element must be left empty. When encountering an empty IE_NET_SERVICES the service discovery will have to proceed to the respective PoA container to carry out the service matching.
- IE_POA_SERVICES introduced in the PoA Container which contains information on services supported at the PoA. For networks offering the same services at all the PoAs, this element must be empty since the information about the services is already specified at the network level.

Figure 3 depicts the example on how the new service information is included in the Information Elements.

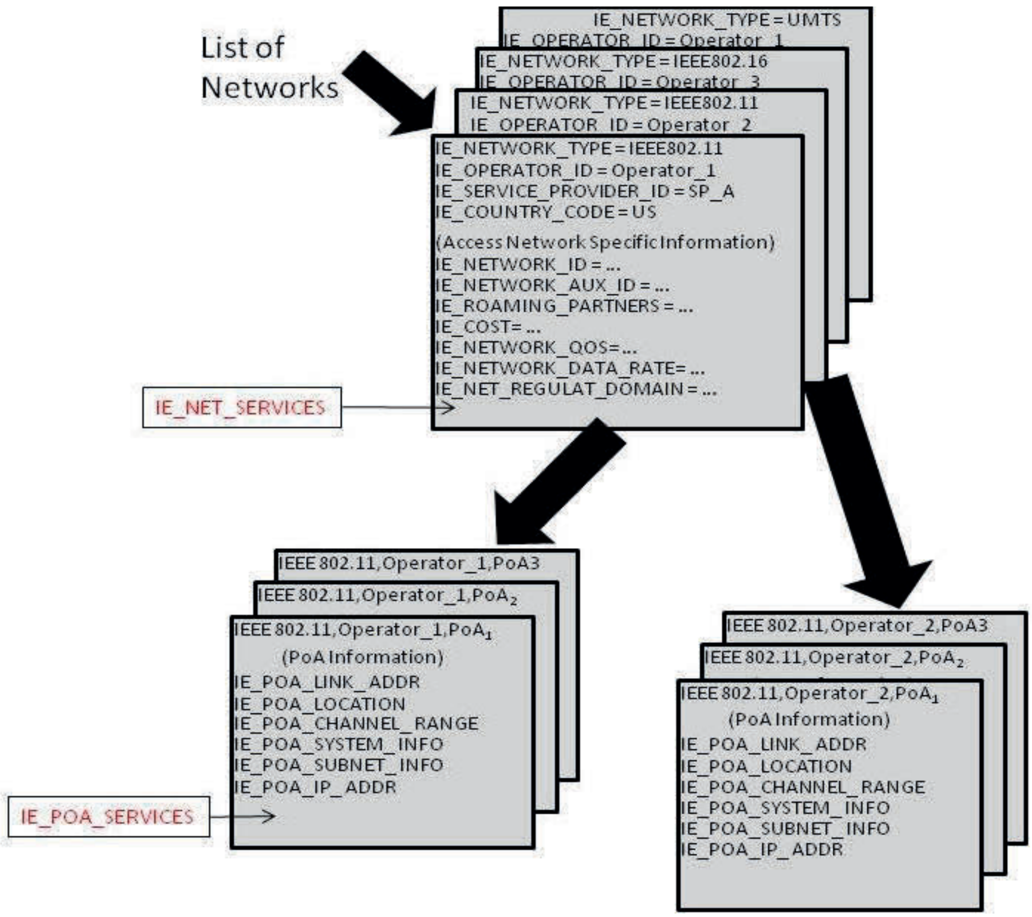

Figure 3. Information elements with service information

7. IMPLEMENTATION OF SERVICE DISCOVERY

In this section the service discovery development environment is described. The user interface is explained and four use scenarios are given to illustrate the effectiveness of the proposed service discovery and matching procedure.

7.1 Mobile Service Discovery Testbed

The service discovery application is developed in the Java^{TM} (G. James, et al 2005) language using the Android Software Development Kit (SDK) which provides the tools and

APIs necessary to begin developing applications on the Android platform. Android (Android developers 2010) is a software stack for mobile devices that includes an operating system, middleware and key applications. Android platform is chosen because it provides access to a wide range of useful libraries and tools that can be used to build rich applications.

Eclipse Helios (Eclipse 2001) is used for the IDE and Android plugin for Eclipse is also needed for the system development. The developed application is flexible and capable of handling all the request variants from the Mobile Node. Android emulator is used for the testing purpose. The application can be installed and run in any types of mobile phones with Android 2.3 operating system and later.

7.2 System User Interface

Figure 4 shows the icon of the service discovery in the Android emulator. Figure 5 depicts the service discovery user interface offering three options to find service as follows:

4. by Service Name - the EquivalenceClass semantic will be used and all the similar services with different names and languages will be return.
5. by Service Type - all the subtype services under the same ParentType will be return.
6. by Keyword - even the non-exact match services will be return as long as it has something in common to describe about the service (for e.g. Snacks or Cocktail to define Food service).

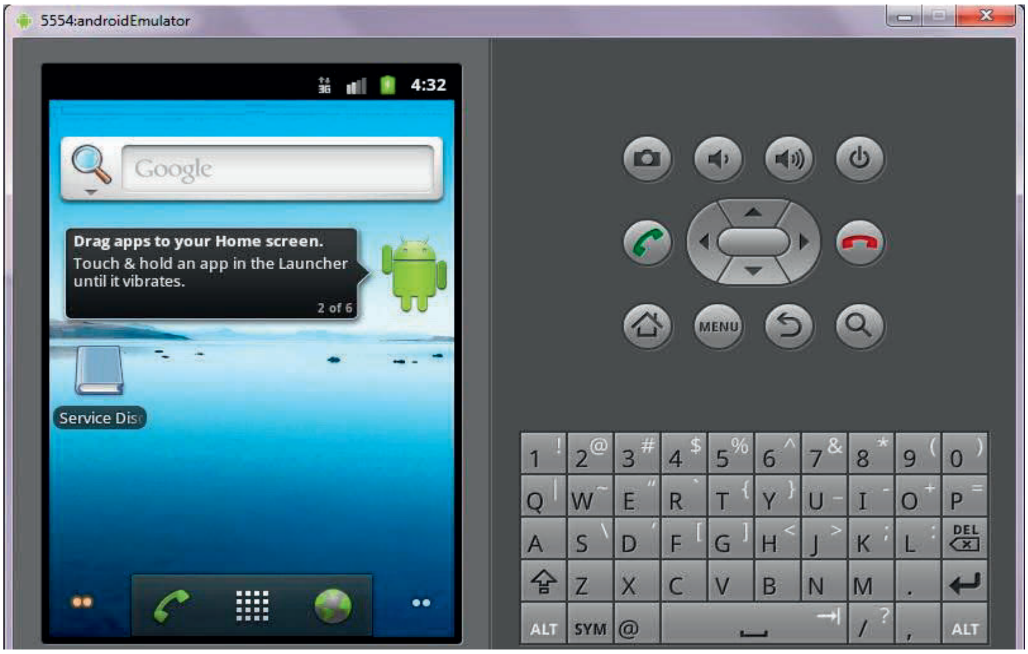

Figure 4. Service discovery icon on Android emulator

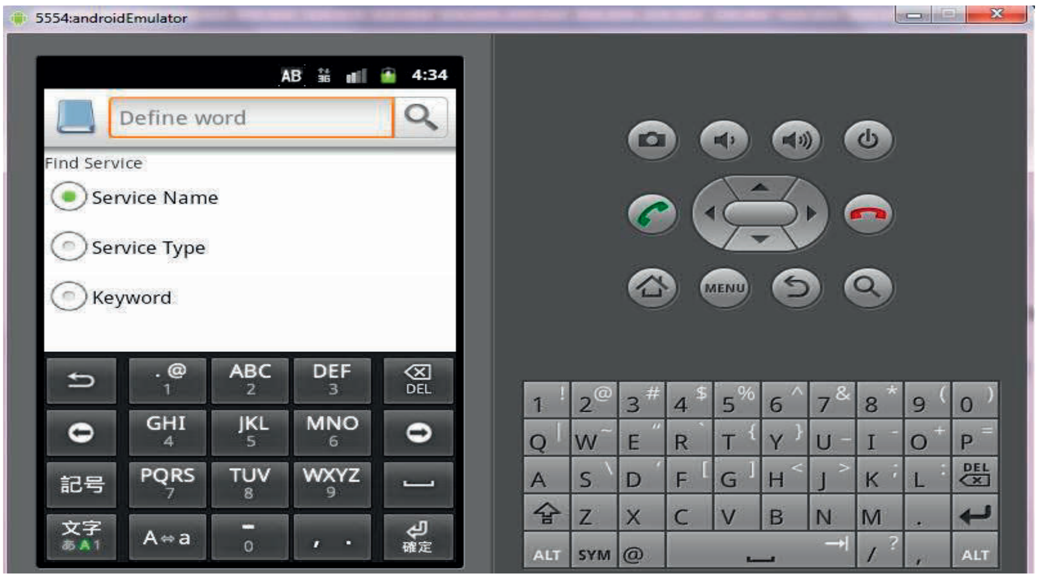

Figure 5. Service discovery user interface

7.3 Use scenarios

Use scenario 1 - Service discovery and matching based on the service name.

The Mobile Node may request for a specific service by specifying its service name. For example, if the Mobile Node requests for a Skype service, only the networks offering Skype service should be returned as depicted in Figure 6.

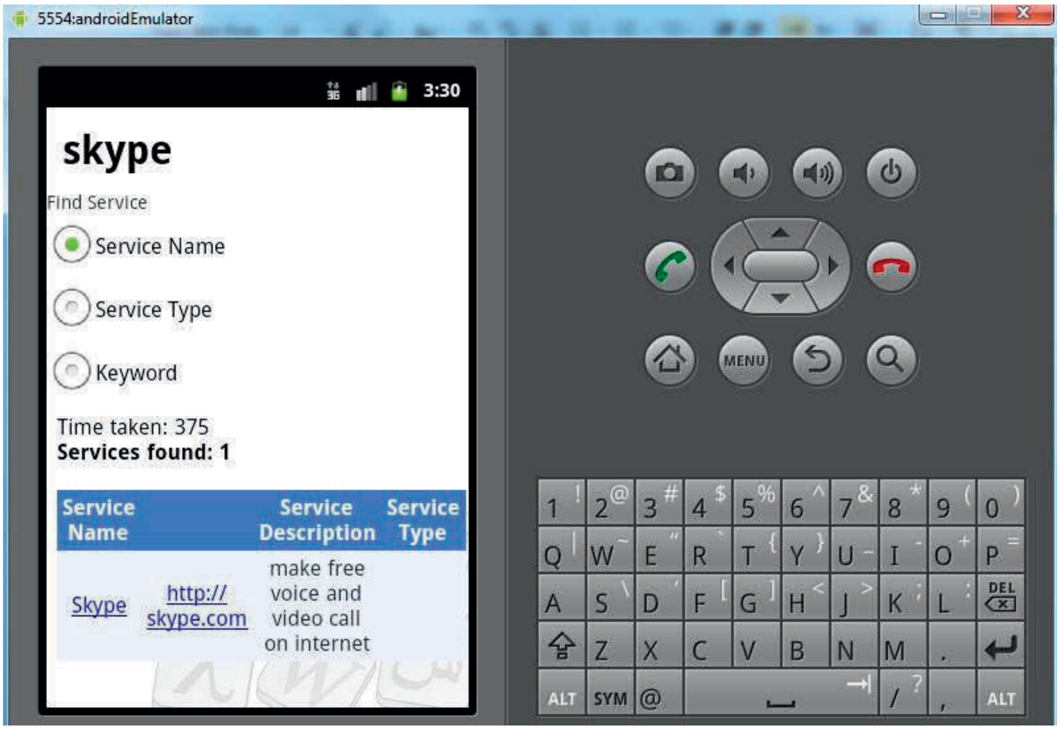

Figure 6. Service discovery and matching based on the service name – Skype example

It is worth noting that a network with a completely different service with the name of Skype can be returned because the service names are not standardized and unique. To avoid this situation the Mobile Node may have to specify the service type in addition to the service name to aid the service discovery. For example, when the Mobile Node requests a service with the service name Book and specify Online Business as service type as illustrated in Figure 7, only the Book service with the service type Online Business is returned.

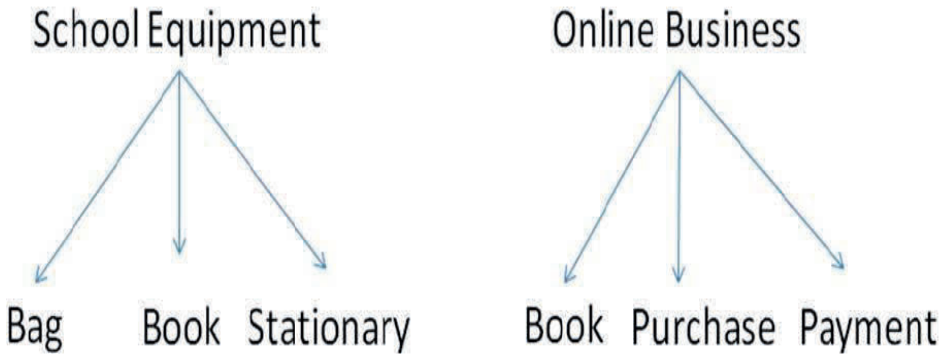

Figure 7. Example of Book service denotes different type of services

Figure 8 shows the example of Mobile Node requests a service with the service name Book without specifying the service type.

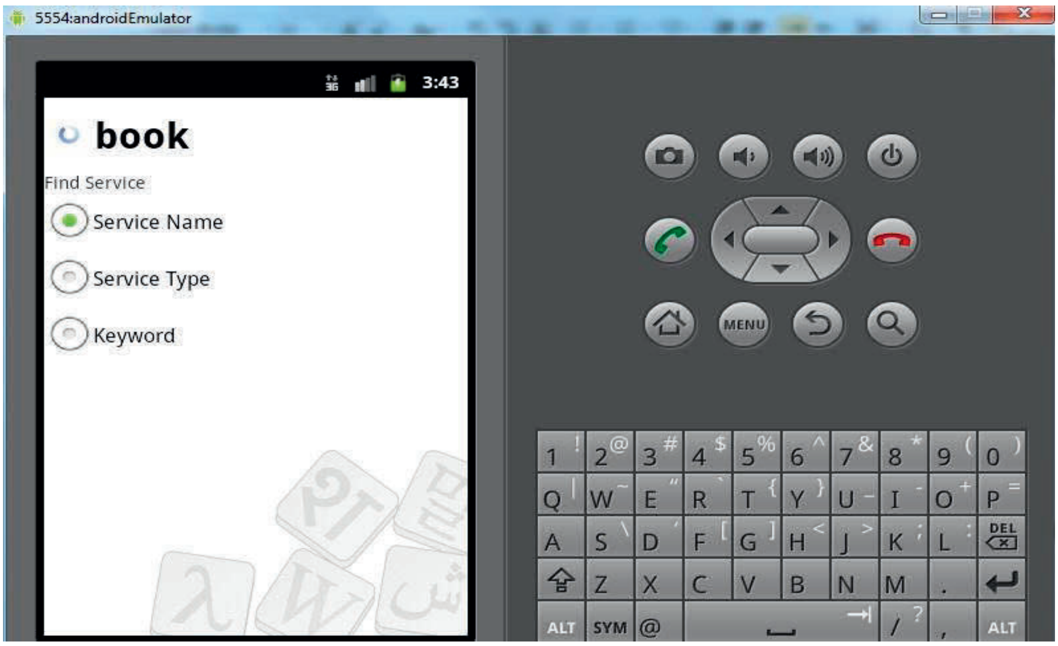

Figure 8. Service discovery and matching based on the service name – Book example

The service discovery will return two completely different book services as the search result as depicted in Figure 9. The first result is referring to a Book service (for e.g. buying online book or information about a book). While the second result of Book service is referring to a Reservation service.

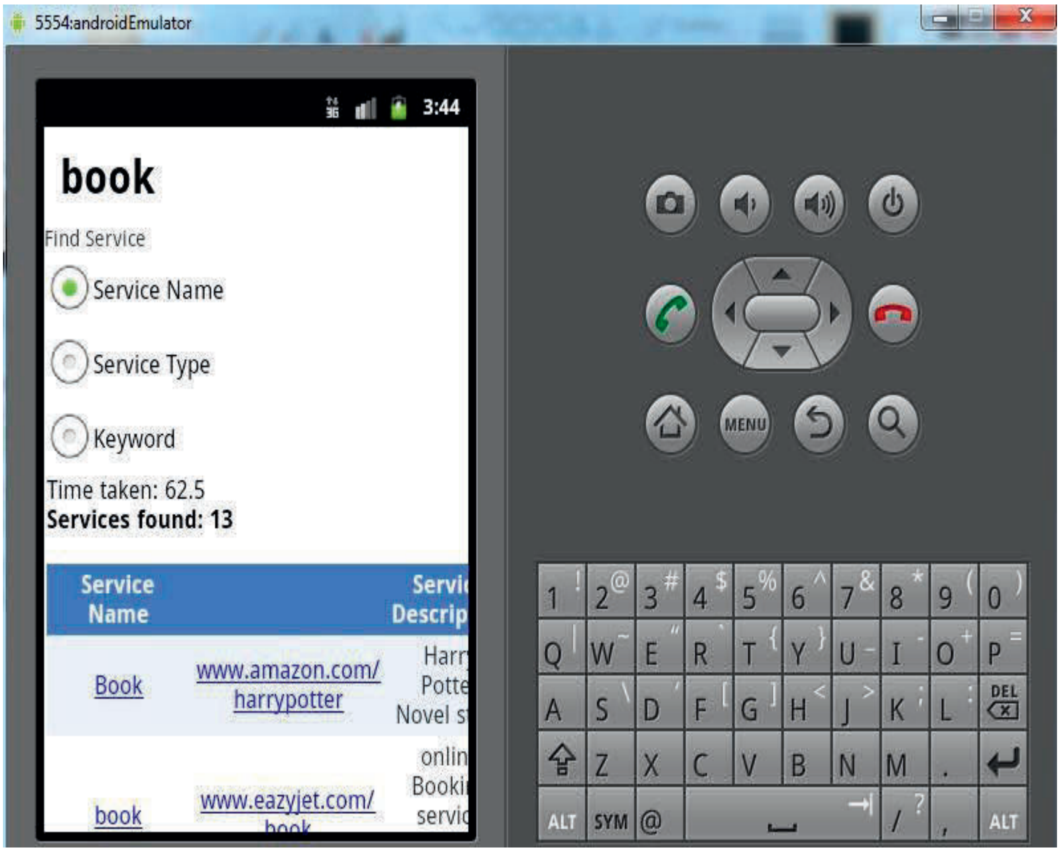

Figure 9. Service discovery and matching based on the service name – Book example (result)

If the Mobile Node specifies only the service type without the service name all the supported services with the same service type will be return. For example, if the Mobile

Node requests for IP telephony service, Skype, VoIP and WiPhone, which has IP Telephony as same service type will be returned.

Use scenario 2 - Service discovery and matching based on ParentType.

The Mobile Node may also request a service by specifying the ParentType. All the services with the same ParentType will be returned. In this case only partial match is required. For example, if the Mobile Node specifies Telephony as ParentType all the child service types of Telephony e.g. Skype, GSM, VoIP and WiPhone will be returned as illustrated in Figure 10.

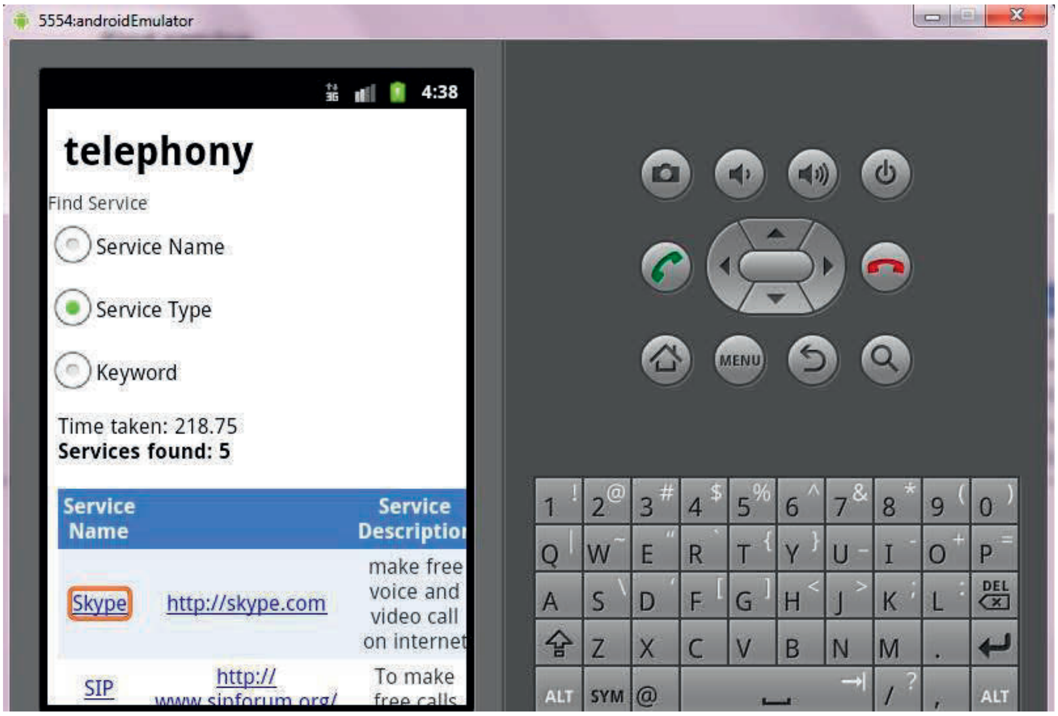

Figure 10. Service discovery and matching based on ParentType

Use scenario 3 - Service discovery and matching based on EquivalenceClass.

The Mobile Node may request a service by specifying the EquivalenceClass. In this case all the services of equivalent service type will be returned. The returned services can have different names, types in different languages. For example, if the Mobile Node requests service by specifying Restoran as EquivalenceClass, Café, Bistro, Restaurant, etc. will be returned as illustrated in Figure 11.

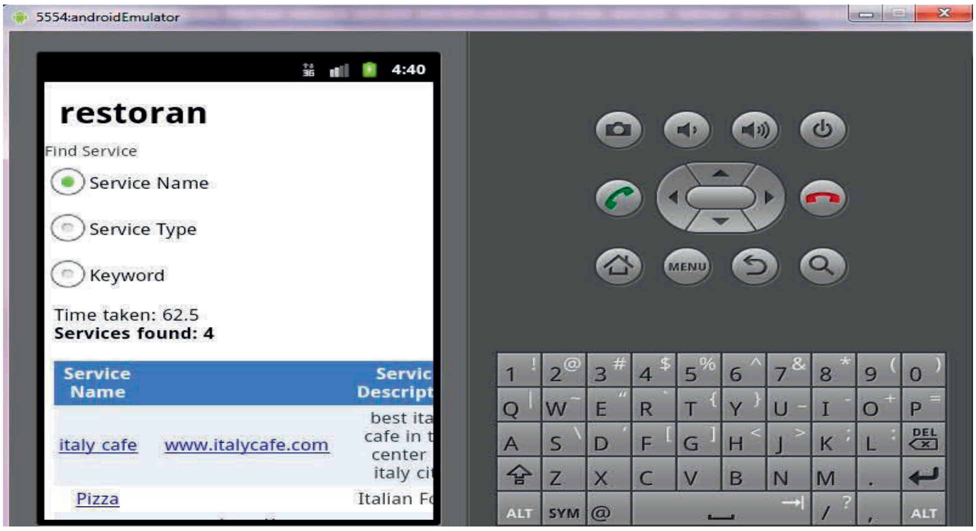

Figure 11. Service discovery and matching based on EquivalenceClass

In addition, the service discovery also supports for finding a service in multiple-languages as shown in Figure 12.

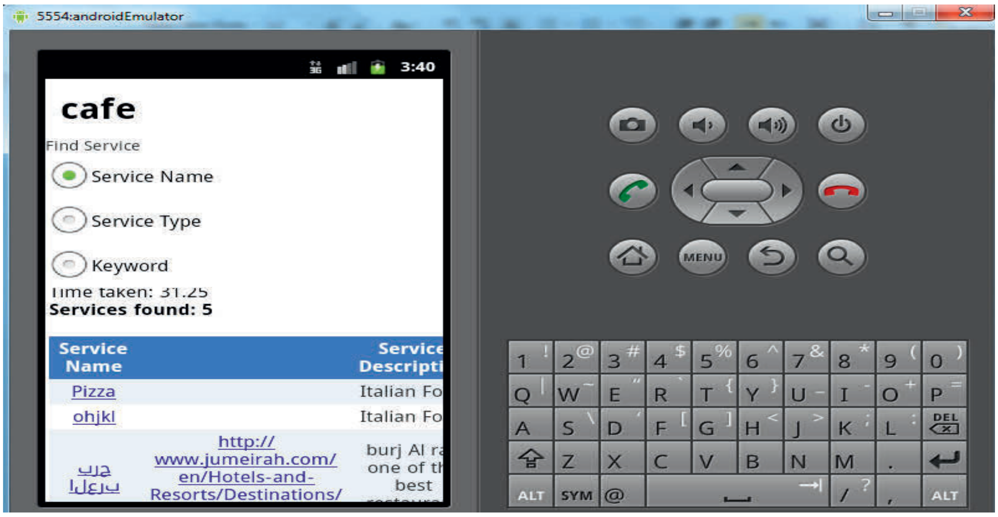

Figure 12. Service discovery and matching based on EquivalenceClass – support for multiple languages

By having this feature the user will not have to worry even though other names returned when he wants to find a service (especially if he is abroad and different languages is used to call a service) because the service discovery system returns only services equivalent to the one requested.

Use scenario 4 - Service discovery and matching based on Keywords.

The Mobile Node can request a service by specifying Keywords. This can be used especially when the user has only a vague idea about the service he wants to request. The service matching the Keywords will be returned. For example, if the Mobile Node wants to find a Food service, it may send Meal, Snacks or Dining as Keywords to request for the Food service as illustrated in Figure 13.

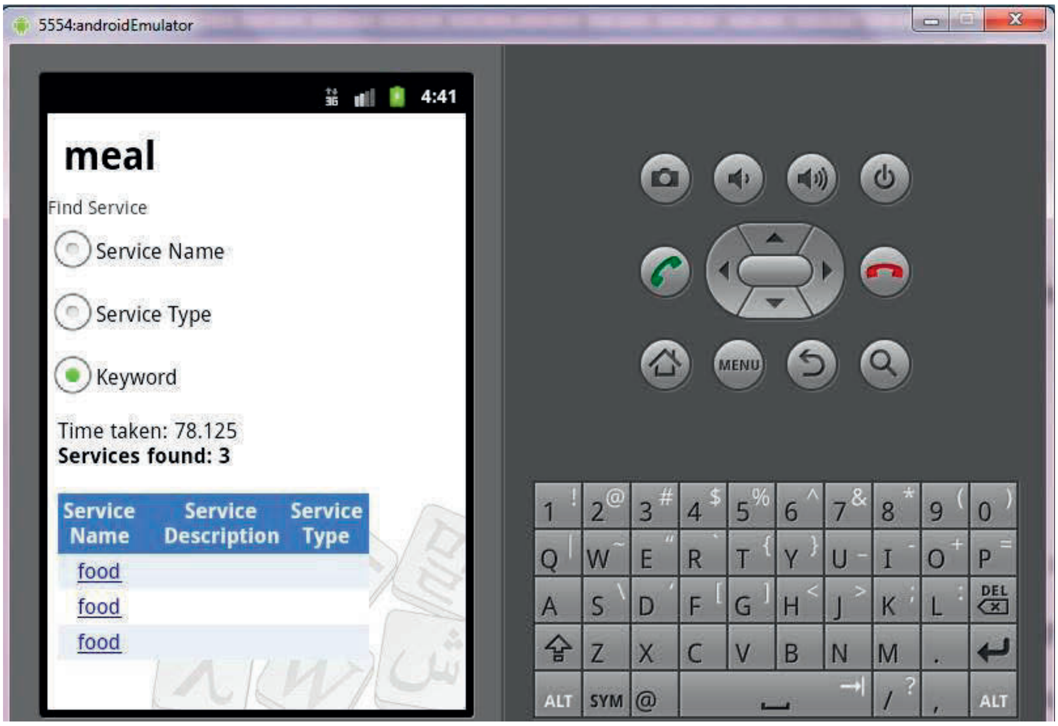

Figure 13. Service discovery and matching based on Keywords.

The time taken in the output screen indicates the processing time (in seconds) to retrieve the search result.

8. CONCLUSION

In this paper, it is explained when and how the IEEE 802.21 MIH fails to support service continuity. To remedy the situation, it is proposed to exchange information about supported services in addition to the network QoS information. The paper also indicates the necessary extensions in the handover procedure and specifies the additional information elements containing service information. A novel service discovery has also been proposed as additional function for the MIH Information Servers. To verify the feasibility of the proposed extension, a simplified and simulated MIH containing only service information and service capability has been successfully implemented (Bashah, et al 2010). The MIH capability including service continuity support on the Mobile Node has also successfully developed on Android and illustrated by few use scenarios in this paper. For further work the proposed extension in the existing IEEE 802.21 MIH Services should be implemented and the service discovery should be installed in any MIH client so that testing and evaluation can be carried out in the real mobile environment.

REFERENCES

Android developers (2009) – http: //developer.android.com/index.html. Accessed October 30, 2010.

B. Wiley, "Distributed Hash Tables, Part 1," *Linux Journal,* Oct 1, 2003 2003.

Bashah, N.S.K, Jørstad, I. & van Do, T, "Service Discovery in Future Open Mobile Environments", Proc. of the 4th International Conference on Digital Society (ICDS2010), IEEE Conference Proceedings, February 2010, ISBN 978-0-7695-3953-9.

Bashah, N.S.K, Kryvinska, N. & van Do, T, "Service Discovery in Ubiquitous Mobile Computing Environment", Proc. of the 12th International Conference on Information Integration and Web-based Applications & Services (IIWAS2010), ACM Publication, November 2010, ISBN 978-1-4503-0440-5.

Bashah, N.S.K, Jørstad, I. & van Do, T, "Enabling Service Continuity on Future Mobile Services", Proc. of the International Symposium on Wireless and Pervasive Computing (ISWPC2009), IEEE Conference Proceedings, February 2009, ISBN 978-1-4244-2965-3.

Bashah, N.S.K, Bhatti, A, Choudhary, I.A, Jørstad, I. & van Do, T, "Service Discovery for Mobile Multi-domain Multi-language Environments", Proc. of the 3rd IEEE International Workshop on Selected Topics in Mobile and Wireless Computing (STMWC2010), IEEE Computer Society, October 2010, ISBN 978-1-4244-7741-8.

D. K. Saehoon Kang, Younghee Lee, Soon J. Hyun, Dongman Lee, and Ben Lee, "A Semantic Service Discovery Network for Large-Scale Ubiquitous Computing Environments," *ETRI Journal,* vol. 29, pp. pp.545-558, Oct. 2007 2007.

Eclipse (2001) - http://www.eclipse.org. Accessed January 30, 2010.

G. James, J. Bill and S. Guy, *The Java Language Specification;* Third Edition: Addison Wesley, 2005.

H. H. Ying Zhang, Dong Yang and Hongke Zhang, "A hierarchical and chord-based semantic service discovery system in the universal network," *International Journal of Innovative Computing, Information and Control,* vol. 5, November 2009 2009.

I. Stoica*, et al.*, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *Networking, IEEE/ACM Transactions on,* vol. 11, pp. 17-32, 2003.

IEEE Computer Society, “IEEE Standard for Local and Metworpolitan Area Networks. Part 21: Media Independent Handover Service”, IEEE Std. 802.21^{TM} – 2008. 21 January 2009.

K. Aberer*, et al.*, "Semantic overlay networks," presented at the Proceedings of the 31st international conference on Very large data bases, Trondheim, Norway, 2005.

Khan, Suabyal; Saastamoinen, Jukka; Huusko, Jyrki; Nurmi, Jari; , "Performance evaluation of distributed NoTA applications on multi-core platforms," *Networked Embedded Systems for Enterprise Applications (NESEA), 2011 IEEE 2nd International Conference on* , vol., no., pp.1-8, 8-9 Dec. 2011

L. D. Ngan*, et al.*, "Dynamic Discovery of Complex Constraint-Based Semantic Web Services," in *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, 2011, pp. 51-58.

N. Kryvinska, C. Strauss, P. Zinterhof, “‘Variated Availability’ Approach to the Services Manageable Delivering”, The Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2011), June 30th-July 2nd, 2011, Korean Bible University (KBU), Seoul, Korea, IEEE, pp. 106-113.

O. W. Group, "OWL Web Ontology Language Overview ", ed, 2004.

P. Yan-Bin*, et al.*, "Method of two stages semantic service discovery," in *Machine Learning and Cybernetics, 2009 International Conference on*, 2009, pp. 1534-1538.

Pirr*, et al.*, "ERGOT: A Semantic-Based System for Service Discovery in Distributed Infrastructures," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, 2010, pp. 263-272.

S. Jiang and F. Aagesen, "An Approach to Integrated Semantic Service Discovery Autonomic Networking." vol. 4195, D. Gaïti*, et al.*, Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 159-171.

T. H. a. K. M. Sim, "An Ontology-enhanced cloud service discovery system," presented at the International MultiConference of Engineers and Computer Scientists 2010 (IMECS2010), Hong Kong, 2010.

V. Kaewmarin*, et al.*, "Semantic Web Service Discovery and Integration Using Service Search Crawler," in *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on*, 2008, pp. 884-888.

Z. Zhao*, et al.*, "MEMORY: A Matrix-Based Efficient Semantic Web Service Discovery System," presented at the Proceedings of the 2009 Fifth International Conference on Next Generation Web Services Practices, 2009.

Bibliography

[1] IEEE Computer Society, 2009, "IEEE Standard for Local and Metropolitan Area Networks—Part 21: Media Independent Handover Services", New York, USA, 323, 802.21 21 January 2009.

[2] Object Management Group, 2004, "Naming Service Specification Version 1.3", Formal/04-10-03 October 2004, http://wenku.baidu.com/view/95421306eff9aef8941e0673.html

[3] Tao Guan, Ed Zaluska & David De Roure, "A Semantic Service Matching Middleware for Mobile Devices Discovering Grid Services," in *Advances in Grid and Pervasive Computing*. vol. 5036, Song Wu, Laurence Yang, and Tony Xu, Eds., ed: Springer Berlin / Heidelberg, 2008, ISBN: 3-540-68081-0, pp. 422-433.

[4] OASIS, 2006, "Reference Model for Service Oriented Architecture 1.0", 2 August 2006, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm

[5] Brahmananda Sapkota, Laurentiu Vasiliu, Ioan Toma, Dumitru Roman & Chris Bussler, "Peer-to-Peer Technology Usage in Web Service Discovery and Matchmaking," in *Web Information Systems Engineering – WISE 2005*. vol. 3806, Anne Ngu, Masaru Kitsuregawa, Erich Neuhold, Jen-Yao Chung, and Quan Sheng, Eds., ed: Springer Berlin / Heidelberg, 2005, ISBN: 3-540-30017-1, pp. 418-425.

[6] Ayomi Bandara, Terry Payne, David De Roure & Tim Lewis, "A Semantic Approach for Service Matching in Pervasive Environments," ECSTR-IAM07-006, http://eprints.ecs.soton.ac.uk/14551/1/ECSTR-IAM07-006.pdf, 2007.

[7] ETSI, 1999, "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN) interface; Gb interface Layer 1", France, 9, ETSI TS 101 298 V7.0.1 (1999-08), August 1999.

[8] 3GPP, 2000, "Technical Specification Group Core Network; NAS Functions related to Mobile Station (MS) in idle mode", FRANCE, 31, 3GPP TS 23.122 V3.1.1 (2000-2), February 2000.

[9] 3GPP, 2008, "Technical Specification Group Radio Access Network; High Speed Packet Access (HSPA) evolution; Frequency Division Duplex (FDD) (Release 7)", 59, 3GPP TR 25.999 V7.1.0 (2008-03), March 2008.

[10] 3GPP, 2008, "3GPP system to Wireless Local Area Network (WLAN) interworking", 3GPP TS 23.234 V7.7.0, June 2008.

[11] 3GPP, 2009, "LTE; Technical Specifications and Technical Reports relating to an Evolved Packet System (EPS) based 3GPP system", 18, 3GPP TS 21.201 version 8.0.0 Release 8, March 2009.

[12] Do Van Thuan, Ivar Jørstad, Boning Feng, Tore Jønvik & Do Van Thanh, "Service Continuity for IMS converged services," in *10th International Conference on Intelligence in Next Generation Networks (ICIN 2008)*, 20^{th}-23^{rd} Oct 2008, ISBN: 978-1-60702-868-0, pp. 6.

[13] *Dialogic*. Available: www.dialogic.com, 31st October 2011.

[14] UPnP, 2011, "UPnP Standardized DCP (SDCP) CallManagement:1 Service For UPnP version 1.0", 312, March 22, 2011.

[15] ETSI, 2002, "Digital cellular telecommunications system (Phase 2+); Man-machine Interface (MMI) of the Mobile Station (MS)", France, 32, ETSI TS 100 907 V7.1.1 (2002-06), June 2002.

[16] Alan R. Hevner, Salvatore T. March, Jinsoo Park & Sudha Ram."Design Science in Information Systems Research". *MIS Quartely* [Research Essay]. (28) 1: 75-105 (2004).

[17] Martin Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition*. Boston, MA: Addison-Wesley Professional, 2004, ISBN: 0321193687.

[18] Nor Shahniza Kamal Bashah, Atif Bhatti, Imran Aslam Choudhary, Ivar Jorstad & Do van Thanh, "Service Discovery for mobile multi-domain multilanguage environments," *IEEE 6th International Conference on* in *Wireless and Mobile Computing, Networking and Communications 2010 (WiMob2010),* Niagara Falls, Canada, 11-13th Oct. 2010, ISBN: 97814244-77432, pp. 675-682.

[19] Ivar Jørstad, "Future Mobile Service Architectures – Service Personalisation and Continuity in Future Mobile Services," PhD Department of Telematics, Norwegian University of Science and Technology, Trondheim, ISBN: 82-471-8152-5, 338, 2006.

[20] IETF, 2002, "RFC 3261: Session Initiation Protocol (SIP)", 203, June 2002, http://www.ietf.org/rfc/rfc3261.txt

[21] 3GPP, 2006, "IP Multimedia Subsystem; Stage 2 Release 5", 3GPP TS 23.228 v5.15.0.

[22] 3GPP, 2005, "3GPP Technical Specification Group Services and System Aspects; Voice Call Continuity between CS and IMS (Release 7) ", 3GPP TR 23.806 V7.0.0 (2005-12).

[23] 3GPP, 2007, "3GPP Technical Specification Group Services and System Aspects; Voice Call Continuity (VCC) between Circuit Switched (CS) and IP Multimedia Subsystem (IMS); Stage 2 ", 3GPP TS 23.206 V7.5.0.

[24] ETSI, 2006, "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Terminology", 12, ETSI TR 180 000 V1.1.1 (2006-02).

[25] ETSI, 2006, "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Services Capabilities, Requirements and strategic direction for NGN services", 95, ETSI TR 181 003 V1.1.1 (2006-05).

[26] W3C Working Group, 2008, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", 26 November 2008, http://www.w3.org/tr/rec-xml/

[27] W3C Working Group, 2004, "Resource Description Framework (RDF): Concepts and Abstract syntax", 10 February 2004, http://www.w3.org/tr/rdf-concepts/

[28] OWL Working Group, 2004, "*OWL Web Ontology Language Overview"* Available: http://www.w3.org/TR/2004/REC-owl-features-20040210/, 30 November 2007

[29] Vladimir Rasin & Steven Knudsen, 2001, "Technical White Paper: The Evolution of Jini Technology in Telematics", Available: www.sun.com/software/jini

[30] UPnP Forum, 2008, "UPnP Device Architecture Version 1.1", 136, 15 October 2008, http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

[31] Salutation Consortium, 1999, "Salutation Architecture Specification Version 2.0C – Part 1", http://systems.cs.colorado.edu/grunwald/MobileComputing/Papers/Salutation/Sa20e1a21.pdf

[32] W3C Working Group, 2004, "Web Services Architecture", 11 February 2004, http://www.w3.org/TR/ws-arch/

[33] Shanshan Jiang & Finn Aarve Aagesen, "An Approach to Integrated Semantic Service Discovery Autonomic Networking." vol. 4195, Dominique Gaïti, Guy Pujolle, Ehab Al-Shaer, Ken Calvert, Simon Dobson, Guy Leduc, and Olli Martikainen, Eds., ed: Springer Berlin / Heidelberg, 2006, ISBN: 978-3-540-45891-3, pp. 159-171.

[34] Giuseppe Pirro, Paolo Trunfio, Domenico Talia, Paolo Missier & Carole Goble, "ERGOT: A Semantic-Based System for Service Discovery in Distributed Infrastructures," in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010),* Melbourne, Australia, 17-20th May 2010, ISBN: 978-1-4244-6987-1, pp. 263-272.

[35] Brandon Wiley, "Distributed Hash Tables, Part 1," *Linux Journal, October* 2003, Issue 114, ISSN: 1075-3583, http://www.linuxjournal.com/article/6797

[36] Karl Aberer & Philippe Cudre-Mauroux, "Semantic overlay networks," in *Proceedings of the 31st international conference on Very large data bases*, Trondheim, Norway, 2005, ISBN: 1-59593-154-6, pp. 1367-1367.

[37] Le Duy Ngan, Lim Yuan Jie & Rajaraman Kanagasabai, "Dynamic Discovery of Complex Constraint-Based Semantic Web Services," in *Fifth IEEE International*

Conference on Semantic Computing (ICSC 2011), Palo Alto, CA, 18th-21st Sept. 2011, ISBN: 978-1-4577-1648-5, pp. 51-58.

[38] Saehoon Kang, Kim Daewong, Lee Younghee, Soon J. Hyun, Lee Dongman & Lee Ben, "A Semantic Service Discovery Network for Large-Scale Ubiquitous Computing Environments," *ETRI Journal,* vol. 29, no. 5, Oct. 2007, ISSN: 12256463, pp. 545-558, http://dx.doi.org/10.4218/etrij.07.0106.0281.

[39] Taekgyeong Han & Kwang Mong Sim, "An Ontology-enhanced cloud service discovery system," Lecture Notes in Engineering and Computer Science, vol. 2180/1, 2010, ISSN: 20780958, pp. 644 – 649.

[40] Ying Zhang, Houkuan Huang, Dong Yang & Hongke Zhang, "A hierarchical and chord-based semantic service discovery system in the universal network," *International Journal of Innovative Computing, Information and Control,* vol. 5, Number 11(A), November 2009, ISSN: 1349-4198, pp. 3745 – 3753.

[41] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek & Hari Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on Networking,* vol. 11, 2003, ISSN: 1063-6692, pp. 17-32.

[42] Zhuo Zhao, Dianfu Ma, Jing Li & Xianyang Qu, "MEMORY: A Matrix-Based Efficient Semantic Web Service Discovery System," Proceedings of the Fifth International Conference on Next Generation Web Services Practices, Prague, 9-11th September 2009, ISBN: 978-0-7695-3821-1, pp. 29–36.

[43] Viriya Kaewmarin, Ngamnij Arch-int & Somjit Arch-int, "Semantic Web Service Discovery and Integration Using Service Search Crawler," in *International*

Conference on Computational Intelligence for Modelling Control & Automation, Vienna, Austria*,* 10-12th Dec. 2008, ISBN: 978-0-7695-3514-2, pp. 884-888.

[44] Yan-Bin Peng, Zhi-Jun Zheng, Ji Gao, Xue-Qin Jiang & Jie-Qing Ai, "Method of two stages semantic service discovery," in *International Conference on Machine Learning and Cybernetics,* Baoding, 12-15 July 2009, ISBN: 978-1-4244-3702-3, pp. 1534-1538.

[45] ISO/IEC, 2005, "Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2 ", Switzerland, 423, ISO/IEC 19501:2005(E), 1 April 2005.

[46] Nor Shahniza Kamal Bashah, Ivar Jorstad & Do van Thanh, "Service Discovery in Future Open Mobile Environments," *Fourth International Conference on Digital Society 2010 (ICDS 2010),* 10-16 Feb 2010, St. Maarten, Netherlands Antilles, 978-0-7695-3953-9, pp. 47-53.

[47] Android Developers. *Android Developers*. Available: http://developer.android.com/sdk/index.html, 26 November 2010.

[48] Ivar Jørstad & Do van Thanh, "Personalisation and Continuity of Mobile Services," in *Mobile Multimedia: Communication Engineering Perspective*, Ismail Khalil Ibrahim, Ed., ed: Nova Science Publishers, Inc., 2006, ISBN: 1-60021-207-7, pp. 59-84.

[49] Nor Shahniza Kamal Bashah, Natalia Kryvinska & Do van Thanh, "Service Discovery in Ubiquitous Mobile Computing Environment," 8*th International Conference on Advances in Mobile Computng and Mutimedia (MoMM 2010)*, Paris, France, 8-10th Nov. 2010, ISBN: 978-1-4503-0440-5, pp. 763-767.

[50] Nor Shahniza Kamal Bashah, Ivar Jorstad & Do van Thanh, "Enabling Service Continuity on Future Mobile Services," *4th International Symposium on Wireless Pervasive Computing 2009 (ISWPC 2009)*, Melbourne, Australia, 11-13th Feb. 2009, ISBN: 978-1-4244-2966-0/09, pp. 1-7.

[51] DAI Yafei & YAN Hongfei, "Name Services," in *Index of /~course/cs501/2005/ppt*, ed, 2005.

[52] Bryan Alexander Ford, "UIA: A Global Connectivity Architecture for Mobile Personal Devices," PhD, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 207, 2008.

[53] E. Mena, V. Kashyap, A. Illarramendi, & A. Sheth, (1998), "Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure", Formal Ontology in Information Systems (1998), Publisher: IOS Press, Pages: 269–283.

[54] Nor Shahniza Kamal Bashah, Ivar Jorstad & Do van Thanh, "Service Naming in Future Mobile Environments," *International Conference on Parallel Processing Workshops 2009 (ICPPW2009)*, Vienna, Austria, 22-25th September 2009, ISBN: 978-0-7695-3803-7, pp. 256-261.

[55] W3C Working Group, 2007, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", http://www.w3.org/TR/soap12-part1/

[56] Fritz Onion, *Essential ASP .Net with Examples in C#*. Boston, MA: Addison Wesley, 2004, ISBN: 0201760401.

[57] Richard Blum, *C#TM Network Programming*. Alameda, CA: John Wiley, 2003, ISBN: 0782141765.

[58] *OwlDotNetApi*. Available: http://users.skynet.be/bpellens/OwlDotNetApi/documentation.html., 2nd February 2010.

[59] Holger Knublauch, Ray W. Fergerson, Natalya F. Noy & Mark A. Musen, "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications," in *The Semantic Web – ISWC 2004*. vol. 3298, Sheila A. McIlraith, Dimitris Plexousakis, & Frank van Harmelen, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 229-243.

[60] James Gosling, Bill Joy, Guy Steele & Gilad Bracha, "*The Java LanguageTM Specification*", Third Edition ed.: Addison Wesley, 2005, ISBN: 0321246780.

[61] *Eclipse*. Available: www.eclipse.org, 18th March 2010.

[62] Jared Stein: Survey of IEEE802.21 Media Independent Handover Services - http://userfs.cec.wustl.edu/~jws2/mihs/index.html

[63] Christian Bettstetter & Christoph Renner, "A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol", 6th EUNICE Open European Summer School: Innovative Internet Applications, 2000.

[64] Reaz Ahmed, Raouf Boutaba, Fernando Cuervo, Youssef Iraqi, Tianshu Li, Noura Limam, Jin Xiao & Joanna Ziembicki, "Service Naming in Large-Scale and Multi-Domain Networks", IEEE Communications Surveys & Tutorials, Third Quarter 2005, Volume 7, No.3, ISSN: 1553-877X, pp. 38-54.

[65] Takahiro Kawamura, Jacques-Albert De Blasio, Tetsuo Hasegawa, Massimo Paolucci & Katia Sycara, "Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry", Service-Oriented Computing –

ICSOC2003, Springer Berlin/Heidelberg Publisher, Lecture Notes in Computer Science, Volume 2910, 2003, ISBN: 978-3-540-20681-1, pp 208-224.

[66] Bart Orriëns, Jian Yang & Mike P. Papazoglou, "Model Driven Service Composition", Service-Oriented Computing - ICSOC 2003, Springer Berlin/Heidelberg Publisher, ISBN 978-3-540-20681, pp 75-90.

[67] Dipanjan Chakraborty, Anupam Joshi, Yelena Yasha & Tim Finin, "Toward Distributed Service Discovery in Pervasive Computing Environments", IEEE Journal of Transactions on Mobile Computing. IEEE Computer Society, Vol. 5. No. 2, 1st February 2006, ISSN: 1536-1233, pp. 97-112.

[68] Vijay Kumar, 2006, "Mobile Database Systems", John Wiley & Sons Inc. Publication, ISBN-13: 9780471467922.

[69] John Daintith, 2008, "A Dictionary of Computing", 6th Edition, Oxford University Press., ISBN: 978-0-19-923400-4.

[70] Ivar Jørstad, Schahram Dustdar & Do van Thanh, "Service Continuity and Personalisation in Future Mobile Services", Proceedings of the 10th International IFIP Workshop on Advances in Fixed and Mobile Networks (EUNICE 2004), Tampere, Finland, 14-16th June 2004, ISBN: 952-15-1187-7, pp. 128-134.

[71] Ivar Jørstad, Do van Thanh & Schahram Dustdar, "An analysis of service continuity in mobile services", Proceedings of the 2nd International Workshop on Distributed and Mobile Collaboration (DMC 2004), IEEE Computer Society Press, University of Modena and Reggio Emilia, Italy, 14-16th June 2004, ISBN: 0-7695-2183-5, pp. 121-126.

[72] Ivar Jørstad, Do van Thanh & Schahram Dustdar, “A Service Continuity Layer for Mobile Services”, Proceedings of the IEEE WCNC Wireless Communications and Networking Conference 2005, New Orleans, USA, 13-17th Mar 2005, ISBN: 0-7803-8967-0, pp. 2300-2305.

[73] Subayal Khan, Jukka Saastamoinen, Jyrki Huusko & Jari Nurmi, "Performance evaluation of distributed NoTA applications on multi-core platforms," *IEEE 2nd International Conference on Networked Embedded Systems for Enterprise Applications (NESEA2011)* , Fremantle, WA, 8-9th Dec. 2011, ISBN: 978-1-4673-0495-5, pp.1-8.

[74] Natalia Kryvinska, Christine Strauss & Peter Zinterhof, “’Variated Availability’ Approach to the Services Manageable Delivering”, The Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2011), Korean Bible University (KBU), Seoul, Korea, June 30th-July 2nd, 2011, ISBN: 978-1-61284-733-7, pp. 106-113.

[75] Marco Aiello, Christian Platzer, Florian Rosenberg, Huy Tran, Martin Vasko & Schahram Dustdar, "Web Service Indexing for Efficient Retrieval and Composition," Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC-EEE '06), San Francisco, CA, 26-29th June 2006, ISBN: 0-7695-2511-3, pp. 63-63.

[76] Josef Noll & Mohammad Chowdhury, "5G: Service Continuity in Heterogeneous Environments," *Wireless Personal Communications,* Volume 57, Number 3 (2011), ISSN: 0929-6212, pp. 413-429, DOI: 10.1007/s11277-010-0077-6 1-17

[77] Cesar Caceres, Alberto Fernandez, Sascha Ossowski & Matteo Vasirani, "Agent-Based Semantic Service Discovery for Healthcare: An Organizational Approach," *Intelligent Systems, IEEE,* Vol. 21, Number 6 (2006), ISSN: 1541-1672, pp. 11-20.

[78] Henning Schulzrinne & Elin Wedlund, "Application-layer mobility using SIP," *ACM SIGMOBILE Mobile Computing and Communications Review,* Vol. 4, Number 3 (2000), ISSN: 1559-1662, pp. 47-57.

[79] Tao Gu, H. C. Qian, J. K. Yao & H. K. Pung, "An architecture for flexible service discovery in OCTOPUS," *Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN 2003),* 20th-22nd Oct. 2003, ISBN: 0-7803-7945-4, pp. 291-296.

[80] Lei Zhang, Jinping Ma & Song Zhang, "Automatic Discovery of Semantic Manufacturing Grid Services," Proceedings of the Third International Conference on Semantics, Knowledge and Grid, Shan Xi, 29th-31st October 2007, ISBN: 0-7695-3007-9, pp. 430-433.

[81] Christos Doulkeridis & Michalis Vazirgiannis, "CASD: Management of a context-aware service directory," *Pervasive and Mobile Computing,* Vol. 4, Number 5 (2008), ISSN: 1574-1192, pp. 737-754, http://www.sciencedirect.com/science/article/B7MF1-4SFXKCT-1/2/8c59d794988158679f1fce775a09dac5

[82] Tara Ali Yahiya, Kaouthar Sethom & Guy Pujolle, "A Case Study: IEEE 802.21 Framework Design for Service Continuity across WLAN and WMAN," in *IFIP International Conference on Wireless and Optical Communications Networks, 2007 (WOCN '07),* Singapore, 2nd-4th July 2007, ISBN: 1-4244-1005-3, pp. 1-5.

[83] Atif Alamri, Mohamad Eid & Abdulmotaleb El Saddik, "Classification of the state-of-the-art dynamic web services composition techniques," International Journal of Web and Grid Services, Volume 2, No. 2 (2006), ISSN: 1741-1106, 2006, pp. 148-166.

[84] Dimitrios Kourtesis & Iraklis Paraskakis, "Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery," in *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science (LNCS), Vol. 5021, 2008, Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, Eds., ed: Springer Berlin / Heidelberg, ISBN: 978-3-540-68233-2, pp. 614-628 .

[85] Peter Haase, Jeen Broekstra, Andreas Eberhart & Raphael Volz, "A Comparison of RDF Query Languages," Proceedings of the Third International *Semantic Web Conference (ISWC 2004), V*ol. 3298 (2004), Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, Eds., ed: Springer Berlin / Heidelberg, 2004502-517, pp. 502-517, doi:10.1007/978-3-540-30475-3_35

[86] Christian Frank & Holger Karl, "Consistency challenges of service discovery in mobile ad hoc networks," Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM 2004), Venice, Italy, ISBN:1-58113-953-5, pp. 105-114, doi:10.1145/1023663.1023684

[87] Pierre-Guillaume Raverdy & Valerie Issarny, "Context-Aware Service Discovery in Heterogeneous Networks," Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM 2005), 13-16 June 2005, ISBN: 0-7695-2342-0-01, pp. 478-480.

[88] Maaz Rehan, Muhammad Yousaf, Amir Qayyum & Shahzad Malik, "A Cross-Layer User Centric Vertical Handover Decision Approach Based on MIH Local

Triggers," in *Wireless and Mobile Networking*, IFIP Advances in Information and Communication Technology, Vol. 308 (2009), Jozef Wozniak, Jerzy Konorski, Ryszard Katulski, and Andrzej Pach, Eds., ed: Springer Boston, ISBN: 978-3-642-03840-2, p. 359.

[89] E. del Val, "Decentralized Semantic Service Discovery based on Homophily for Self-Adaptive Service-Oriented MAS" 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, ISBN: 978-0-9826571-7-1, pp. 1347-1348.

[90] Kim Kyung-ah, Hong Jin Young, Choi Young-Soo, Kim Yong Ho & Lee Seong-Choon, "Design and Implementation of IEEE 802.21 Framework for WIBRO-HSDPA Seamless Mobility," in *International Conference on New Trends in Information and Service Science 2009 (NISS '09),* Beijing, China, 30th June-2nd July 2009, ISBN: 978-0-7695-3687-3, pp. 1284-1290.

[91] Masateru Minami, Hiroyuki Morikawa & Tomonori Aoyama, "The design of naming-based service composition system for ubiquitous computing applications," in *International Symposium on Applications and the Internet Workshops 2004 (SAINT 2004 Workshops),* 26-30th January 2004, ISBN: 0-7695-2050-2, pp. 304-312.

[92] DLNA. *Digital Living Network Allaiance (DLNA)*. Available: www.dlna.org/en/industry/home, 21st September 2007.

[93] José Viterbo F., Markus Endler & Vagner Sacramento, "Discovering services with restricted location scope in ubiquitous environments," Proceedings of the 5th International Workshop on Middleware for Pervasive and Ad-hoc Computing: held at the ACM/IFIP/USENIX 8th International Middleware Conference, Newport

Beach, California, 2007, Newport Beach California, ISBN: 978-1-59593-930-2, pp. 55-60.

[94] Kaarthik Sivashanmugam, Kunal Verma & Amit Sheth, "Discovery of Web Services in a Federated Registry Environment," Proceedings of the IEEE International Conference on Web Services (ICWS 2004), ISBN: 0-7695-2167-3, pp. 270-278.

[95] IETF, 1987, "RFC 1035: Domain Names – Implementation and Specification," 55, November 1987, http://www.ietf.org/rfc/rfc1035.txt

[96] Tomas Vitvar, Maciej Zaremba & Matthew Moran, "Dynamic Service Discovery Through Meta-interactions with Service Providers," Proceedings of the 4th European Conference on The Semantic Web: Research and Applications (ESWC'07), Innsbruck, Austria, ISBN: 978-3-540-72666-1, pp. 84-98.

[97] Marco Crasso, Alejandro Zunino & Marcelo Campo, "Easy web service discovery: A query-by-example approach," *Science of Computer Programming,* Vol. 71, Issue 2, ISSN: 0167-6423, pp. 144-164, 2008, http://www.sciencedirect.com/science/article/B6V17-4S03RB3-1/2/3faa855acf62d7ec2837380b86d142f8

[98] Lin Yi-Bing, Chen Whai-En & Gan Chai-Hien, "Effective VoIP call routing in WLAN and cellular integration," *Communications Letters, IEEE,* Vol. 9, Issue 10, ISSN: 1089-7798, 2005, pp. 874-876.

[99] Pierre-Guillaume Raverdy, Oriana Riva, Agnes de La Chapelle, Rafik Chibout & Valerie Issarny, "Efficient Context-aware Service Discovery in Multi-Protocol Pervasive Environments," in *7th International Conference on Mobile Data Management (MDM 2006),* 10-12th May 2006, ISBN: 0-7695-2526-1, pp. 3-3.

[100] Sonia Ben Mokhtar, Anupam Kaul, Nikolaos Georgantas & Valérie Issarny, "Efficient Semantic Service Discovery in Pervasive Computing Environments," Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware (Middleware 2006), Vol. 4290, Maarten van Steen and Michi Henning, Eds., ed: Springer Berlin / Heidelberg, pp. 240-259.

[101] Ying Wang, Yun Zhou, Jun Yuan & Ping Zhang, "An Enhanced Media Independent Handover Framework for Heterogeneous Networks," in *IEEE Vehicular Technology Conference (VTC Spring 2008)*, Singapore, 11-14th May 2008, ISBN: 978-1-4244-1645-5, pp. 2306-2310.

[102] Weibin Zhao & Henning Schulzrinne, "Enhancing Service Location Protocol for efficiency, scalability and advanced discovery," *Journal of Systems and Software – Special Issue: Software Engineering,* Vol. 75, Issue 1-2, ISSN: 0164-1212, pp. 193-204, 2005, http://www.sciencedirect.com/science/article/B6V0N-4CJVJ2C-1/2/c23764258351720045c25ab4dba5a43a

[103] Alexandros Kaloxylos, George Lampropoulos, Nikos Passas & Lazaros Merakos, "A flexible handover mechanism for seamless service continuity in heterogeneous environments," Journal of Computer Communications, Vol. 29, Issue 6, ISSN: 0140-3664, pp. 717-729, March 2006, http://www.sciencedirect.com/science/article/B6TYP-4H16KXH-3/2/ccca080bdfc632c326fb2ef9f2d137ab

[104] Ciaramita Massimiliano, Hofmann Thomas & Johnson Mark, "Hierarchical semantic classification: Word sense disambiguation with world knowledge," Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), pp. 817-822.

[105] Antonio de la Oliva, Telemaco Melia, Albert Vidal, Carlos J. Bernardos, Ignacio Soto & Albert Banchs, "IEEE 802.21 enabled mobile terminals for optimized WLAN/3G handovers: a case study," ACM *SIGMOBILE Mobile Computing Communications Review,* Vol. 11, Issue 2, ISSN: 1559-1662, pp. 29-40, 2007.

[106] E. C. Popovici, O. Fratu & S. V. Halunga, "An IEEE 802.21-based approach of designing interoperability modules for vertical handover in wireless hybrid access networks," in *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE 2009), Aalborg, Denmark,* 17-20th May 2009, ISBN: 978-1-4244-4066-5, pp. 82-86.

[107] Wonseok Lee, Mun-Seok Kang & Mi-Sook Lim, "Implementation of 802.21 for Seamless Handover Across Heterogeneous Networks," Proceedings of the 10th Asia-Pacific Network Operations and Management Symposium on Managing Next Generation Networks and Services (APNOMS 2007), Vol. 4773, Shingo Ata and Choong Hong, Eds., ed: Springer Berlin / Heidelberg, ISBN: 978-3-540-75475-6, pp. 326-333.

[108] Sanaullah Nazir, Brahmananda Sapkota & Tomas Vitvar, "Improving Web Service Discovery with Personalized Goal," in *Web Information Systems and Technologies*. Vol. 18, Part 3, José Cordeiro, Slimane Hammoudi, and Joaquim Filipe, Eds., ed: Springer Berlin Heidelberg, 2009, ISBN: 978-3-642-01344-7, pp. 266-277.

[109] Magdalena Balazinska, Hari Balakrishnan & David Karger, "INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery," Proceedings of the First International Conference on Pervasive Computing (Pervasive 2002), ISBN: 3-540-44060-7, pp. 195-210.

[110] T. Buburuzan, G. May, T. Melia, J. Modeker & M. Wetterwald, "Integration of Broadcast Technologies with Heterogeneous Networks - An IEEE 802.21 Centric Approach," in *International Conference on Consumer Electronics (ICCE 2007),* Las Vegas, NV, 10-14th Jan. 2007, ISBN: 1-4244-0763-X, pp. 1-2.

[111] Vitor Jesus, Susana Sargento, Miguel Almeida, Daniel Corujo, Rui L. Aguiar, Janusz Gozdecki, Gustavo Carneiro, Albert Banchs & Pablo Yáñez-Mingot, "Integration of mobility and qos in 4g scenarios," Proceedings of the 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2007), Chania, Crete Island, Greece, ISBN: 978-1-59593-806-0, pp. 47-54.

[112] Filippo Cacace & Luca Vollero, "Managing mobility and adaptation in upcoming 802.21 enabled devices," Proceedings of the 4th International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH 2006), Los Angeles, CA, USA, ISBN: 1-59593-470-7, pp. 1-10.

[113] Lee Kyu Min, Choi Kee-Hyun, Her Seung-Phil & Shin Dong-Ryeol, "Matchmaking algorithms to improve dynamic service matching in ubiquitous environments," Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS 2005), ISBN: 0-7695-2296-3, pp. 239-244.

[114] G. Lampropoulos, A. K. Salkintzis & N. Passas, "Media-independent handover for seamless service provision in heterogeneous networks," *IEEE Communications Magazine,* Vol. 46, Issue 1, ISSN: 0163-6804, pp. 64-71, 2008

[115] John Buford, Alan Brown & Mario Kolberg, "Meta Service Discovery," Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2006, ISBN: 0-7695-2520-2, pp. 124.

[116] Kunal Verma, Kaarthik Sivashanmugam, Amit Sheth, Abhijit Patil, Swapna Oundhakar & John Miller, "METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services," *Information Technology and Management Journal,* Vol. 6, Issue 1, ISSN: 1385-951X, pp. 17-39, 2005, http://dx.doi.org/10.1007/s10799-004-7773-4

[117] Vijay Kumar, *Mobile Database Systems*: John Wiley & Sons Inc. Publication, 2006, ISBN: 9780470048283, pages 320.

[118] Andrea Calvagna, Aurelio La Corte & Sabrina Sicari, "Mobility and quality of service across heterogeneous wireless networks," *Computer Networks:* The International Journal of Computer and Telecommunications Networking - Wireless IP through integration of wireless LAN and cellular networks, Vol. 47, Issue 2, ISSN: 1389-1286, pp. 203-217, 2005.

[119] Li Mo, K. Sandrasegaran & T. Tung, "A Multi-Interface Proposal for IEEE 802.21 Media Independent Handover," in *International Conference on the Management of Mobile Business (ICMB 2007)*, Toronto, Ontario, Canada, 9-11th July 2007, ISBN: 0-7695-2803-1, pp. 7-7.

[120] Daniel Schober: Naming Conventions for Ontology Engineering - The European Bioinformatics Institute (EBI) Net Project – Postdoctoral Ontologist – http://www.ebi.ac.uk/net-project

[121] Baek Joo-YoungKim Deok-JinSuh Young-JooHwang Eui-Seok & Chung Young-Don, "Network-Initiated Handover Based on IEEE 802.21 Framework for QoS Service Continuity in UMTS/802.16e Networks," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, 11-14 May 2008, 1550-2252, pp. 2157-2161.

[122] Yeo Bong Yoon & Hee Yong Youn, "A New Service Discovery Scheme Adapting to User Behavior for Ubiquitous Computing," Proceedings of the 2005 International Conference on Computational Science and Its Applications (ICCSA 2005), ISBN: 978-3-540-25862-9, pp. 19-28.

[123] Meriem Kassar, Brigitte Kervella & Guy Pujolle, "An overview of vertical handover decision strategies in heterogeneous wireless networks," *Computer Communications Journal,* Vol. 31, Issue 10, ISSN: 0140-3664, pp. 2607-2620, 2008.

[124] Lee Choonhwa & Helal Sumi, "Protocols for Service Discovery in Dynamic and Mobile Networks," Journal of Computer Research, Vol. 11, No. 1, pp. 1-12, 2002.

[125] Ashutosh Dutta, Subir Das, David Famolari, Yoshihiro Ohba, Kenichi Taniuchi, Victor Fajardo, Rafa Marin Lopez, Toshikazu Kodama & Henning Schulzrinne, "Seamless proactive handover across heterogeneous access networks," *Journal Wireless Personal Communication,* Vol. 43, Issue 3, ISSN: 0929-6212, pp. 837-855, 2007.

[126] Zhang Ying, "Semantic-Based Data and Service Unified Discovery," Proceedings of the Fourth International Joint Conference on *Computational* Sciences and Optimization (CSO 2011), 15-19th April 2011, ISBN: 978-0-7695-4335-2 , pp. 787-791.

[127] Miguel Angel Corella & Pablo Castells, "Semi-automatic Semantic-Based Web Service Classification," Proceedings of the International Conference on Business Process Management Workshops (BPM 2006), Vol. 4103, Johann Eder and Schahram Dustdar, Eds., ed: Springer Berlin / Heidelberg, ISBN: 978-3-540-38444-1, pp. 459-470.

[128] G. G. Richard, III, "Service advertisement and discovery: enabling universal device cooperation," *IEEE Internet Computing Journal*, Vol. 4, ISSN: 1089-7801, pp. 18-26, 2000.

[129] Liljana Gavrilovska & Ramjee Prasad, "Service Discovery," in *Ad Hoc Networking Towards Seamless Communications*, ed: Springer Netherlands, ISBN: 978-1-4020-5066-4, pp. 143-172, 2006.

[130] Alessandro Ordine, Julian F. Gutierrez & Luca Veltri, "SIP roaming solution amongst different WLAN-based service providers," in *3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops, 2006.*, 17th-21st July 2006, ISBN: 1-4244-0498-3, pp. 1-7.

[131] Y. C. Yee, K. N. Choong, Andy L. Y. Low & S. W. Tan, "SIP-based proactive and adaptive mobility management framework for heterogeneous networks," *Journal of Network and Computer Applications,* Vol. 31, Issue 4, ISSN: 1084-8045, pp. 771-792, 2008.

[132] W3C Working Group, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)," ed, 2007.

[133] Lei Li & Ian Horrocks, "A software framework for matchmaking based on semantic web technology," Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, 2003, ISBN: 1-58113-680-3, pp. 331-339.

[134] Mohamrnad. A. Makhzan & Lin Kwei-Jay, "Solutions to a Complete Web Service Discovery and Composition," Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (Ec-EEE 2006), 26-29th June 2006, ISBN: 0-7695-2511-3, pp. 73-73.

[135] Adrian Friday, Nigel Davies, Nat Wallbank, Elaine Catterall & Stephen Pink, "Supporting service discovery, querying and interaction in ubiquitous computing environments," *Wireless Networks – Special Issue: Pervasive Computing and Communication,* Vol. 10, Issue 6, ISSN: 1022-0038, pp. 631-641, 2004.

[136] Tao Gu, Hung Keng Pung & Jian Kang Yao, "Towards a flexible service discovery," *Journal of Network and Computer Applications,* Vol. 28, Issue 3, ISSN: 1084-8045, pp. 233-248, 2005.

[137] S. Grech, H. Haverinen & V. Devarapalli, "Towards Service Continuity in Emerging Heterogeneous Mobile Networks," in *IEEE 63rd Vehicular Technology Conference (VTC 2006-Spring),* Melbourne, Australia, 7-10th May 2006, ISBN: 0-7803-9391-0, pp. 47-51.

[138] Genevieve Bartlett, John Heidemann & Christos Papadopoulos, "Understanding passive and active service discovery," Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC 2007), San Diego, California, USA, ISBN: 978-1-59593-908-1, pp. 57-70, 2007.

[139] Chou Wu, Li Li & Liu Feng, "WIP: Web Service Initiation Protocol for Multimedia and Voice Communication over IP," in *International Conference on Web Services (ICWS '06), Chicago, IL,* 18th-22nd Sept. 2006, ISBN: 0-7695-2669-1, pp. 515-522.