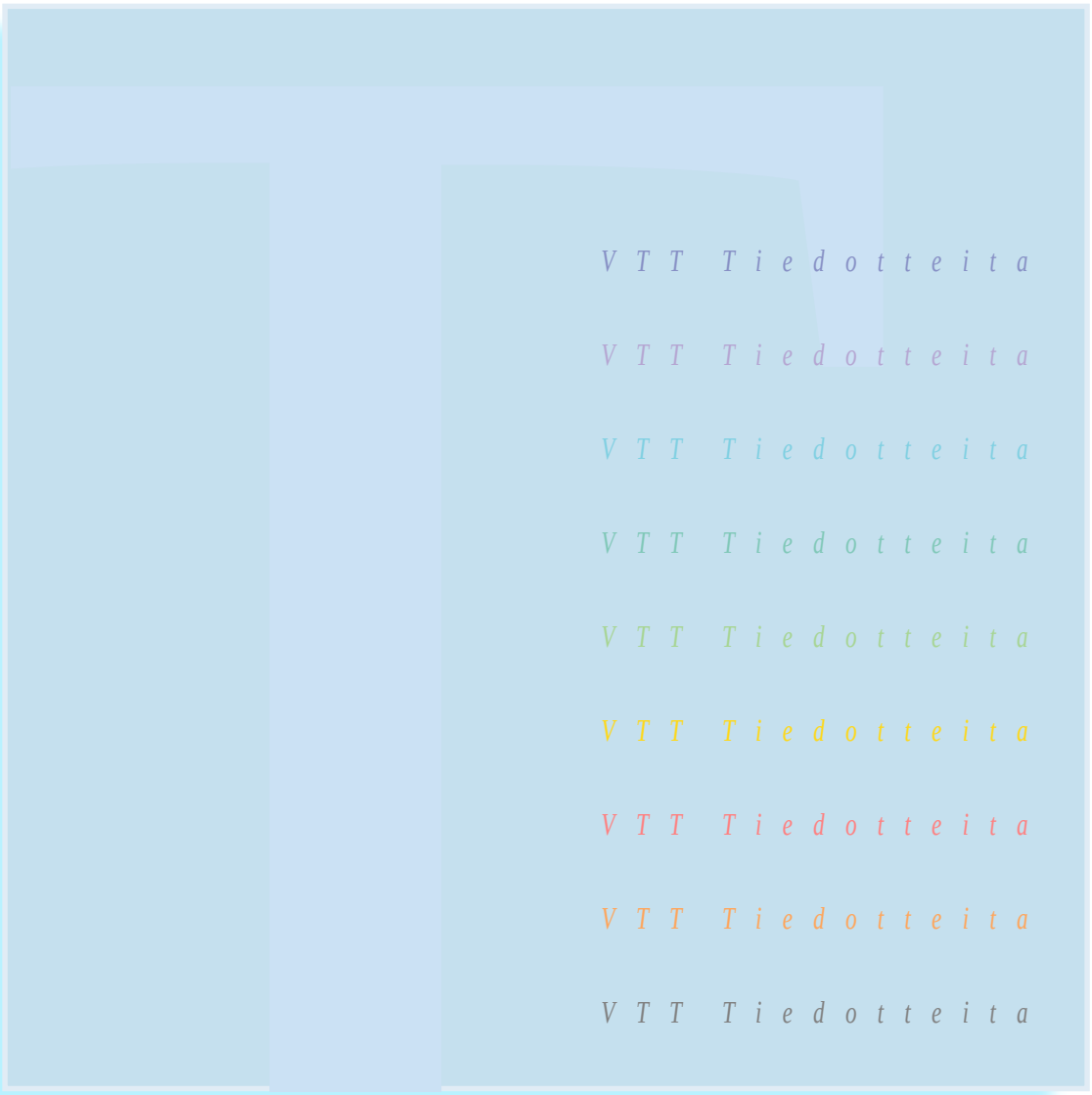
2065

VTT RESEARCH NOTES

VTT TIEDOTTEITA

Markus Sihvonen

A user side framework for Composite Capability / Preference Profile negotiation



TECHNICAL RESEARCH CENTRE OF FINLAND

ESPOO 2000

VTT TIEDOTTEITA – MEDDELANDEN – RESEARCH NOTES 2065

A user side framework for Composite Capability / Preference Profile negotiation

Markus Sihvonen
VTT Electronics

TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 2000

ISBN 951–38–5768–9 (soft back edition)
ISSN 1235–0605 (soft back edition)

ISBN 951–38–5769–7 (URL: http://www.inf.vtt.fi/pdf/)
ISSN 1455–0865 (URL: http://www.inf.vtt.fi/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Kerttu Tirronen

Otamedia Oy 2000

Sihvonen, Markus. A user side framework for Composite Capability / Preference Profile negotiation. Espoo 2000, Technical Research Centre of Finland, VTT Tiedotteita – Meddelanden – Research Notes 2056. 54 p. + app. 4 p.

Keywords Mobile Station Application Execution Environment (MExE), Virtual Home Environment, Open Service Architecture, mobile terminal, CC/PP description

Abstract

The Mobile Station Application Execution Environment (MExE) is a standard, which is aimed at smart mobile terminals and its purpose is to facilitate intelligent network services. It is a part of the Virtual Home Environment, the mobile extension of the Open Service Architecture. Before a mobile terminal (MT) can download MExE applications, it must transfer its Composite Capability / Preference Profile (CC/PP) description to a MExE server. This master thesis proposes strategies for the transfer of CC/PP descriptions to the MExE servers. The utilised research method is a constructive research method that is based on abstract analysis of written material of the studied subject.

The CC/PP description may originate from multiple sources and there must be a clearly defined strategy for transferring it to the MExE server. A MT must be able to manage dynamically its resources in MExE. This thesis has three research problems: what should be the source of the MT's CC/PP description, what is a feasible CC/PP description transfer strategy for MTs and what are the dynamic resource management requirements of the mobile terminal?

The proposed CC/PP strategy suggests fetching the CC/PP description in fragments from servers in the network and from the MT. This considerably improves the efficiency of CC/PP transfer in mobile networks. Changes to the capabilities of the MT can be transmitted when necessary to the MExE server. MExE satisfies the requirements defined by the thesis and it has potential to become a widely used standard.

The following studies will concentrate on designing decision capabilities for a MExE server and a common resource vocabulary for MTs. The designed server will also support dynamic resource maintenance for MTs. A test environment will be constructed, which allows for a closer evaluation of present and future results.

Preface

I wish to thank Mr. Hannu Honka and Mr. Hannu Rytilä from VTT Electronics, for giving me this chance to work in a very interesting project and for giving me strong support while I was working on my Master thesis. Particularly Mr. Hannu Rytilä has been a priceless help in guiding me into a right direction and reviewing periodically the progress of the study. Professor Samuli Saukkonen from the University of Oulu has had an important role of directing me during the writing process and providing in depth instructions about the academic requirements of the Master thesis. I am also thankful to Mr. Johan Plomp from VTT Electronics for allocating time from his busy schedule to review this study. In general, the whole working environment in VTT Electronics in Oulu Finland encourages a young researcher to do the best possible job.

Contents

Definitions

Application:	Services, which are designed using service capability features [16].
CC/PP description:	The device capabilities and user preferences that are described in the CC/PP framework. A CC/PP description is intended to provide information necessary to adapt the content and the content delivery mechanism to best fit the capabilities and preferences of the user and its agent [10].
CC/PP repository:	An application program that maintains CC/PP description [10].
Client:	A program that establishes connections for the purpose of sending requests [5].
Client device:	A mobile device that utilises the services offered by the VHE.
Home Environment:	Is a network that is responsible for overall provision of services to user [16].
Local Service:	A service exclusively provided in the current serving network by a value added service provider [12].
MExE classmark 1:	Is based on WAP and requires limited input and output facilities on the client side. It is designed to provide quick and cheap information access over narrow and slow data connection. It is proposed by WAP forum as a transport protocol for wireless networks. It is based on the Wireless Transaction Protocol (WTP), the Wireless Transport Layer Security and the Wireless Session Protocol (WSP) [14].
MExE classmark 2:	Is based on Personal-Java. It provides and utilises a run-time system requiring more processing, storage, display, and network resources. It also supports more powerful applications and more flexible MMIs. It also includes support for MExE classmark 1 applications. It is designed for Java enabled devices with telephony specific extensions [14].
OSA:	A vendor independent means for the introduction of new services [13].

OSA interface: Standardised interface used by an application to access service capability features [16].

Personal service portfolio

A user defined group of services in the VHE environment. Services can be added and deleted from the portfolio at any moment. A user can have any number of personal service portfolios.

Resource: A network data object or service that can be identified by a URI [5].

Services: Services are made up of different service capability features [16].

Service Capabilities: Bearers defined by parameters, and/or mechanisms needed to realise services. These are within networks and under network control [16].

Service Capability Feature:

Functionality offered by service capabilities that are accessible via the standardised OSA interface [16].

Service Capability Server: Functional Entity providing OSA interfaces towards an application [16].

Service personalisation: Modification of behaviour that may involve the service features or data of a service, within the limitations set by the provider of the service [12].

User agent: The client that initiates a request. They can be browsers, editors, spiders or any other end user tool [5].

User profile: This is a label identifying a combination of one user interface profile, and one user services profile [16].

Value Added Service Provider:

Provides services other than basic telecommunications service for which additional charges may be incurred [12].

Virtual Home Environment:

A concept for personal service environment portability across network boundaries and between terminals [16].

List of symbols

CC/PP	Composite Capability / Preference Profiles.
HTTP	Hypertext Transfer Protocol.
ITEA	Information Technology for European Advancement.
MExE	Mobile Station (application) Execution Environment.
MIME	Multipurpose Internet Mail Extension.
MMI	Man Machine Interface.
MP	Mobile Phone.
MS	Mobile Station.
MT	Mobile Terminal.
OSA	Open Service Architecture.
RDF	Resource Definition Framework.
SCS	Service Capability Server.
UMTS	Universal Mobile Telecommunications System.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
VHE	Virtual Home Environment.
WAP	Wireless Application protocol.
WSP	Wireless Session Protocol.
XML	Extensible Markup Language.

1. Introduction

This study is a part of a larger project, which is called ITEA-VHE (Information Technology for European Advancement – Virtual Home Environment). The ITEA is one of the a EUREKA Projects and its purpose is to strengthen European software and software engineering competences. The total budget of the ITEA is estimated to be 3.2 billion EURO and total effort will be aproximately 20,000 person years. The Project focuses on stimulating and supporting the development of software technology competences for usage by European industry, small, medium and larger sized enterprises alike. The ITEA Project was proposed by ten leading industrial companies, that utilize software technology as a core competence for the creation of their products: Alcatel, Barco, Bull, DaimlerChrysler, ltaltel, Nokia, Philips, Robert Bosch, Siemens and Thomson. The ITEA project will benefit the European ICT sector at large. Funding is envisaged and solicited from Eureka members [18].

The goal of the VHE project is to create a leading role for the European industry in the domain of middleware for end-user terminals with wireless connections and corresponding infrastructure. This is done by developing world-class software technologies for new products that can piggy-back on the fast growing markets for cellular voice communication products and services especially taking into account mobile users and services as well [18].

The consortium is aiming at the definition of middleware software technologies to be used in the application server and in end-user terminals. This establishes virtual home environments (VHE) that let users retain and personalise their services anywhere, and use them at any time in both wireless and wired environments. Beyond the technical level it is within the scope of the project to identify those VHE services attracting end-users and potential providers as well, e.g. electronic picture mail and simplified electronic-commerce via Virtual Home Environments. Participating organisations in the VHE project are Paderborn University, VTT, Orga, Siemens IC C-LAB, Bosch GmbH, Bull, Nokia and Philips [18].

This study is concerned with the transmission of a client device's profile description to a server. A profile description includes information about the hardware and the software manufacturers default settings, modifications of the default settings and user-defined settings. According to the profile description of a client device, a server delivers the requested service or application to the client device. The profile description of a client device and the transmission of it to a server will be discussed more detailed later in this chapter (chapter 1.4). This study aims to determine the suitability of the CC/PP technology for transmitting a client devices' profile information to a server. The problem includes the selection of a CC/PP transfer protocol strategy and proper underlying technology.

The transmission of a client device's profile information is part of larger concepts called the Open Service Architecture (OSA) and the Virtual Home Environment (VHE). The VHE has considerable overlap with OSA but it also includes VHE-specific features. In order to integrate the Open Service Architecture and the Virtual Home Environment, it

is necessary to manage the resources of all client devices that participate in the system. One solution is to deploy CC/PP negotiation framework. The Mobile Station Application Execution Environment is identified by the VHE as one of the supporting mechanisms for the VHE. Before getting deeper into the CC/PP transfer protocol, the OSA, the VHE, and the MExE will be introduced in more detail.

1.1 The Open Service Architecture

The Open Service Architecture is meant to be a highly flexible environment, which enables the implementation of a priori unknown end user services and applications. The application may be e.g. a small downloadable Java program that enables a user to access a bank account by using his MT. Once the service session is finished, the application will be deleted from the MT's memory. It is particularly an architecture that enables applications to utilise network capabilities. Applications will access the network via a specially designed OSA interface [16].

The network functionality is accessed via different Service Capability Servers (SCSs). OSA binds applications and service capabilities provided by the underlying network together. Application developers have access to these capabilities when designing new applications. The OSA interface is a basic building block for application developers to rapidly construct new applications. It is also possible to combine different features of different SCSs' whenever needed. The main goal of OSA is to provide an extensible and scalable architecture that allows addition of new service capability features and SCSs in future releases of UMTS with minimum impact for the applications that deploy the OSA interface [16]

The OSA is constructed from three different parts, which are the application portion, the framework portion, and the service capability servers as illustrated in figure 1. Applications can be implemented into one or more application servers. The framework provides to applications the basic mechanism that allows them to utilise service capabilities in the network. In order for an application to use the network functionality, which is provided by the service capability servers, authentication between the application and the framework is needed. Once the authentication has occurred, the discovery service enables the application to find out what service capability features are provided by the service capability servers in the framework. The service capability servers provide for applications the service capability features that are abstractions of the underlying network functionality. It is usual that more than one service capability server will offer similar functionality [16].

.

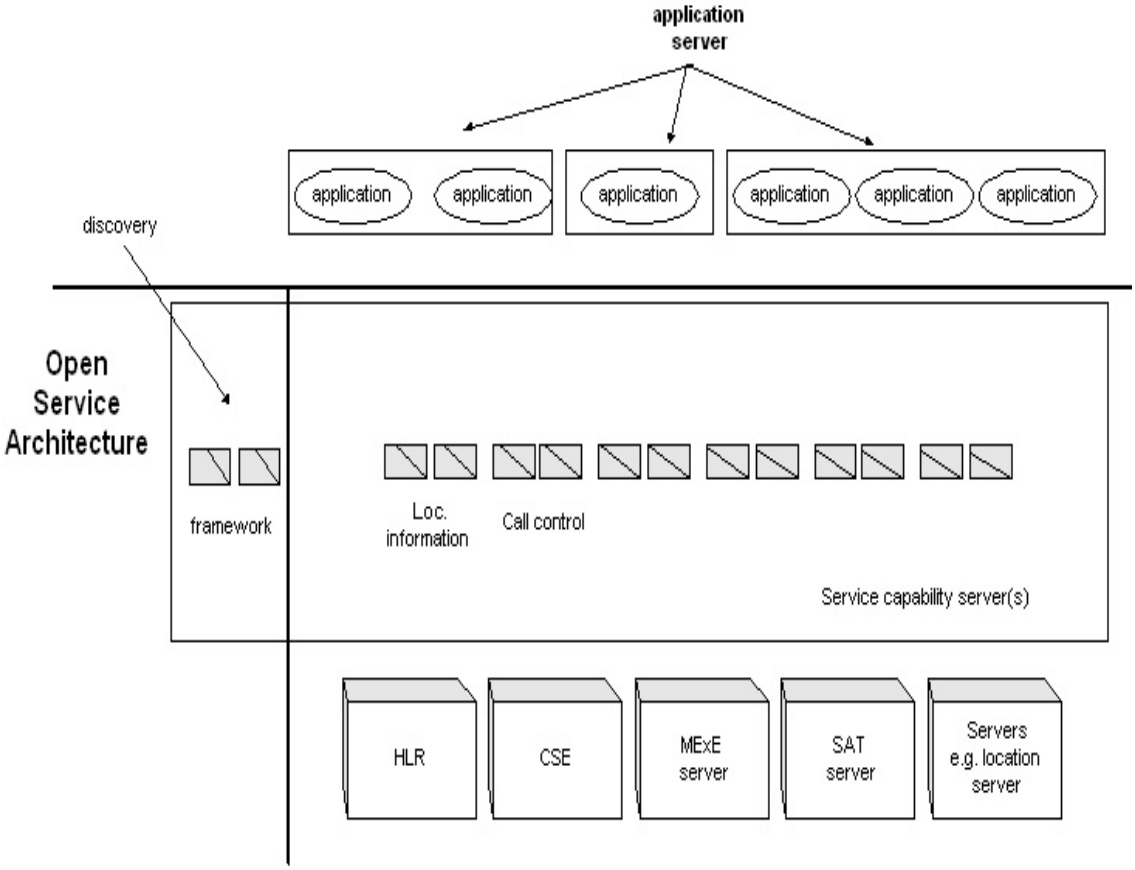


Figure 1. Overview of Open Service Architecture.

The OSA service capability features are specified in terms of a number of interface classes and their methods. The interface classes are divided into two groups as illustrated in figure 1. The groups are the framework interface classes and the service interface classes. The framework interface classes describe the methods on the framework and the service interface classes illustrate the methods on the service capability servers [16].

1.2 The Virtual Home Environment

The Virtual Home Environment (VHE) is a concept for personalised service portability across network boundaries and between terminals. The main idea is that users are consistently presented with the same personalised features regardless of the network and the terminal the user is deploying [12].

The roles and components involved in the realisation of the VHE are following; the home environment, one or more unique identifiers; one user; one or more terminals; one or more serving network operator; one subscription and possibly one or more value added service providers. This is illustrated in figure 2. The main characteristics of the VHE are a user ability to manage a service portfolio and possibility to access value-added services via any value added service provider [12].

The purpose of the VHE is to provide services to a user in a consistent manner, which also means that a user can have multiple user profiles. A user can have a different user profile for different physical environments. For example, a user can have a different user profile for office purposes and for an automobile environment. A user's VHE is a combination of services, profiles and personalization information that compose a user's personal service portfolio. The VHE enables the creation of services by providing accesses to service capabilities by means of a standardised interface [12].

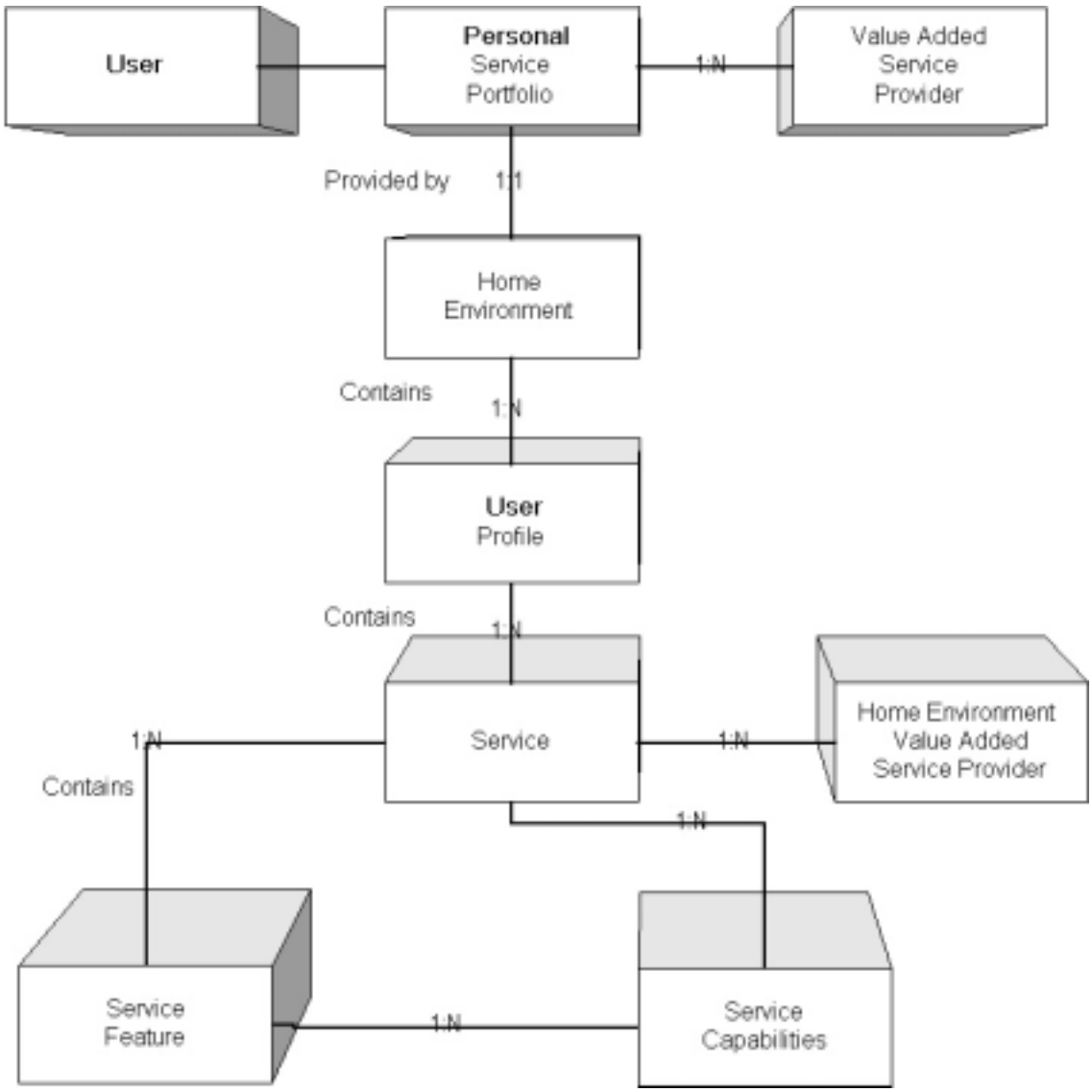


Figure 2. Role of Components involved in Realisation of VHE.

In the VHE, a user will have the possibility to manage services and their appearance. A user can also personalise the services and the user interface. He can access services from any network or terminal that has the required capabilities. A user will have the capability to access new services at any given moment or deactivate old services. It is possible to modify a user profile from any location. It is possible to find new local services in a secure manner and indicate which subscription charges are applicable at any given moment. A user can select any particular user profile and recover a mobile station's resident user profile information when needed [15].

1.3 The Mobile Station Application Execution Environment

The Mobile Station Application Execution Environment (MExE) is a wireless protocol, which is embedded into smart mobile terminals. It is designed to be a full application execution environment for mobile terminals. The purpose of MExE is facilitating intelligent network services and providing e.g. sophisticated and intelligent customer menus. MExE supports many man-machine interfaces such as voice recognition, icons and softkeys. There are plans for incorporating phone location services into MExE. The aim of MExE is providing a comprehensive and standardised environment for mobile phones for executing operator or service provider-specific applications. MExE is aimed for next generation powerful mobile terminals, but it can also be incorporated into today's regular mobile phones [20].

The architectural model in figure 3 shows an example of how a GSM network uses standardised transport mechanisms to transfer MExE services between the MS and the MExE service environment. The same architectural model can also be applied in next generation mobile networks. The MExE service environment can include several service nodes each providing MExE services that can be transferred to the MS using standard Internet protocols. The MExE service environment can include a proxy server to translate content defined in standard Internet protocols into their wireless optimised derivatives [17].

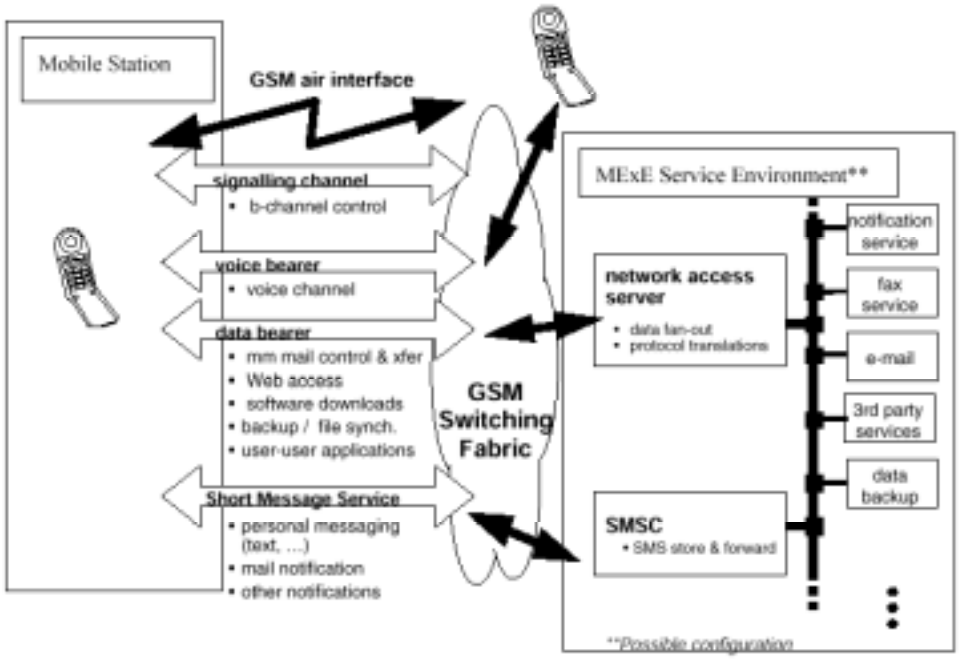


Figure 3. Generic MExE architecture.

1.4 Profile transfer scenarios

The purpose of this chapter is to introduce two separate VHE user scenarios in order to illustrate further the research area. They are an observation camera scenario and a fax-printing scenario. More information about these scenarios is available in the appendix.

1.4.1 Observation camera scenario

This scenario introduces the remote use of an observation camera via the Internet. The observation camera is controlled by a mobile terminal. Detailed background information about this user scenario can be found in appendix 1 and the configuration is illustrated in figure 4. Before a user can gain full control of the Observation Camera by using the mobile terminal, he needs to upload the required application software from his Home Desktop / Server. Furthermore before any event can take place, the mobile terminal must communicate its current profile information to the home desktop / server. This is to confirm that the mobile terminal has the required capabilities, both in hardware and software, to execute the application and to determine the current status of the user defined preferences.

This scenario is fully fictional and all the information presented in it is completely imaginary. In the scenario the mobile terminal has the following profile information:

Hardware platform

- Vendor = Nokia
- Model = 9150
- Type = MT (Mobile Terminal)
- Screen = Yes
- Screen size = 800*600*24
- Keyboard = Yes
- Keyboard type = Mobile
- CPU = PPC
- Bluetooth = Yes
- Memory = 32MB
- Speakers = Yes

Software platform

- Operating system = EPOC1.0
- HTML version = 4.0
- Java Script version = 4.0

- WAP version = 1.0
- WML Script = 1.0
- Sound = ON
- Images = ON

User preferences

- Language = English

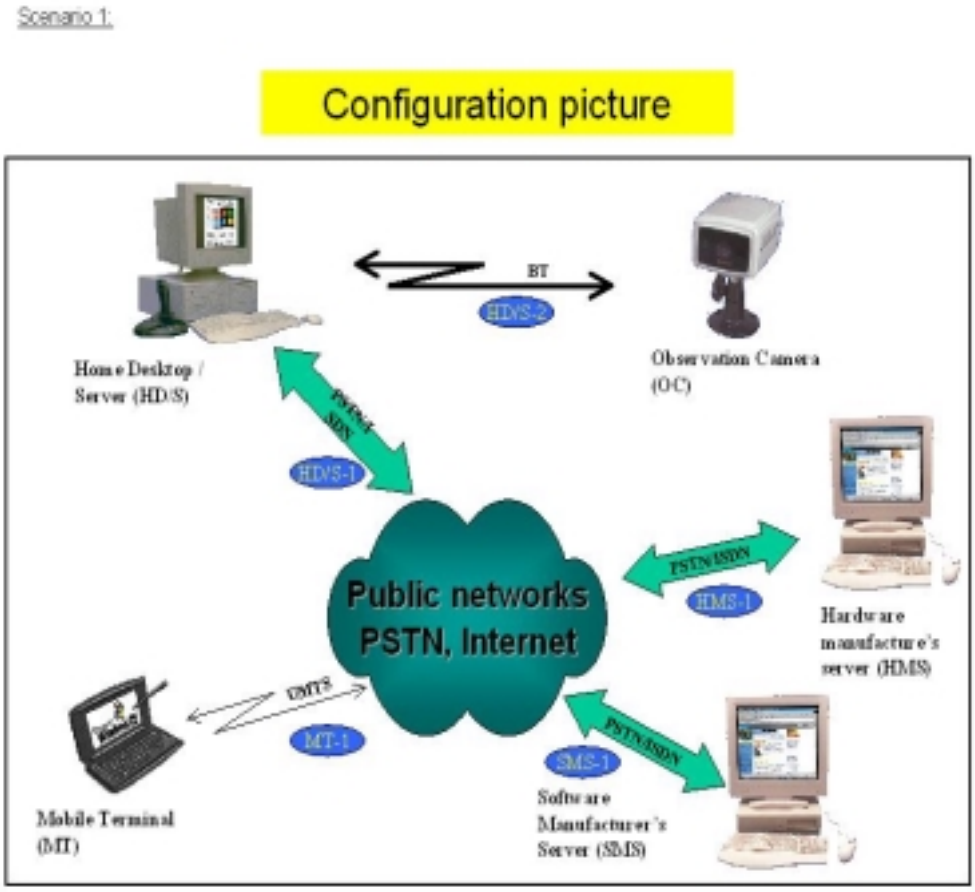


Figure 4. Configuration picture for scenario 1.

Neither the hardware nor the software manufacturer's default Composite Capability / Preference Profiles description settings are downloaded from the MT to the Server, since the default settings are absolute values that can not be changed by the user in any circumstances. The Mobile Terminal downloads only the user preferences and absolute URIs of the hardware and software default settings to the Server. Usually default settings of any given piece of hardware or software can be found from its manufacturer's maintained server, but it is not necessary. The home server in the scenario gets the default settings from the URI locations and only the user preferences from the MT.

It is also debatable how much of the user preferences the MT should download during the first connection. Basically the MT can transfer all information at once, transfer nothing at all or transfer something between these two extreme choices. The disadvantage of mobile devices and wireless networks is that wireless networks will always be relatively slower than wired networks, which must be taken into account when communicating profile information. When the manufacturers' default settings are downloaded from a server that is part of a wired network, the first part of the problem is solved. The second part of the problem can be solved by having the MT to download only the bare minimum CC/PP description during the first transfer. If the Server needs additional information about the MT in order to provide the requested service, the server must request that particular information. It just needs to be decided what is the bare minimum CC/PP description that must be transferred at the first transfer session? This discussion will be left for the later part of the thesis.

Fax printing scenario

More detailed background information for the user scenario is provided in appendix 2 and the configuration is illustrated in figure 5. In this scenario a visitor's MP detects VTT's Local Area Network immediately after it has arrived into Blue Tooth's operational range and requests access to the network's services. Since VTT's LAN administrator allows visitors to access some services it provides, the MP is granted a visitor access status in the LAN after which it can use predefined services.

This scenario is fully fictional and all the information presented in it is completely imaginary. In the scenario, the Mobile Phone has following CC/PP description:

Hardware platform

- Vendor = Nokia
- Model = 10150
- Type = MP (Mobile Phone)
- Screen = Yes
- Screen size = 100*50*12
- Keyboard = Yes

- Keyboard type = Mobile Phone
- CPU = PPC
- Bluetooth = Yes
- Memory = 16MB
- Speakers = Yes

Software platform

- Operating system = EPOC1.0
- HTML version = 4.0
- Java Script version = 4.0
- WAP version = 1.0
- WML Script = 1.0
- Sound = ON
- Images = ON

User preferences

- Language = Finnish

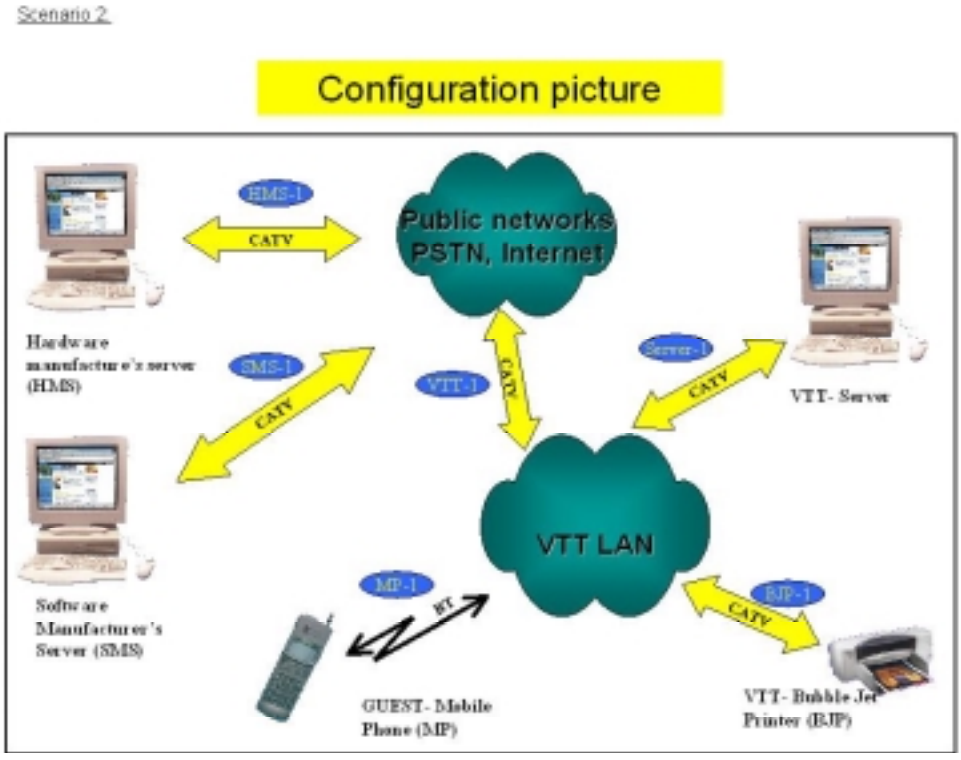


Figure 5. Configuration picture for scenario 2.

In the scenario the VTT-Server needs to have some information about the MP the first time when it requests access to the VTT LAN. But how much information should be transferred during this first connection? Again there are three alternatives, which are the full CC/PP description, just the access request, or something between the first two choices. This paper will address this issue in later chapters.

The same issue in transmitting the manufacturers' default settings resurfaces in this scenario as in the first scenario. Where should the MP's default CC/PP descriptions be located and how they should be transferred to the VTT- Server? Would the best strategy for MP be just to transfer the URIs of the default settings of the hardware and the software to the VTT-Server, or should it itself transfer the full CC/PP description, including default settings, to the VTT-Server? This problem will also be discussed in later chapters in this thesis.

1.5 Research problems and methods

Although previously discussed scenarios are different from each other the same unsolved questions of how to manage the client device profile information arise from both examples. The research problems are:

1. What should be the source of the client device's default profile information?

2. What is feasible profile information transfer strategy of the client device?

3. What are the dynamic resource management requirements of the client device?

The utilised research method is a constructive research method that is based on abstract analysis of written material of the studied subject [7]. Chapter 2 introduces the principal technologies that are discussed when analysing the research problems. The technology issues that are involved in describing, transferring, and managing a client device's profile description are discussed in chapter 3. Chapter 4 introduces a protocol strategy for a client device, which is used when it transmits its profile description to a server. A framework for transferring a profile description is discussed in chapter 5 and conclusions of the study are introduced in chapter 6.

2. Technology overview

This chapter briefly introduces the main issues of the principal technologies discussed in this study. They are the Composite Capability / Preference Profile, the HTTP 1.1 extension framework and the Resource Definition Format.

2.1 The Composite Capability / Preference Profile

The CC/PP framework is a mechanism for describing the capabilities and preferences associated with users and user agents. Information about user agents includes the hardware platform, system software, applications and user preferences. The user agent capabilities and preferences can be thought of as metadata or properties and descriptions of the user agent hardware and software. The CC/PP descriptions are intended to provide information necessary to adapt the content and the content delivery mechanisms to best fit the capabilities and preferences of the user and its agents. [10]

The CC/PP exchange protocol does not depend on the profile format, which it conveys. Therefore another profile format besides the CC/PP description format could be applied to the CC/PP exchange protocol [10].

The basic requirements for the CC/PP exchange protocol are listed below.

- Transmission of the CC/PP descriptions should be WSP or HTTP/1.1-compatible [5].

- The CC/PP exchange protocol should support an indirect addressing scheme based on RFC2396 [1] for referencing profile information.

- Components used to construct CC/PP descriptions, such as vendor default descriptions, should be independently cacheable.

- The CC/PP exchange protocol should provide a lightweight exchange mechanism that permit clients to avoid re-sending the elements of the CC/PP descriptions that have not changed since the last time the information was transmitted.

- The protocols must be able to use gateways and proxies if they exist [9].

The basic data model for a CC/PP is a collection of tables. In the simplest form each table in the CC/PP is a collection of RDF statements with simple, atomic properties. These tables may be constructed from default settings, persistent local changes or temporary changes made by a user. One extension to the simple table of properties data model is the notion of a separate, subordinate collection of default properties [10].

2.2 The HTTP 1.1 extension framework

The Hypertext Transfer Protocol is an application level protocol for distributed, collaborative, hypermedia information systems. It has been used by the World Wide Web global information initiative since 1990. HTTP 1.1 has been developed since HTTP 1.0 does not sufficiently consider the effects of hierarchical proxies, caching, the need for persistent connections and the need for virtual hosts. The need for the reliable exchange of capability information between client and server has also contributed to the need for HTTP 1.1, since it deploys stricter requirements than HTTP 1.0 in order to ensure secure implementation of its features [5].

HTTP allows an open-ended set of methods and headers that indicate the purpose of a request. It uses Uniform Resource Identifiers for indicating the resource to which a method is to be applied. The HTTP protocol is also used for communicating between user agents and proxies or gateways to other Internet systems [5].

The HTTP protocol's type is request/respond. A request is send to a server by a client in the request method form - URI, protocol version, MIME-like message that contains request modifiers, client information and possible body content over a connection with a server. A server responds by sending a status line, which contains the message protocol version, success or error code, MIME-like message that contains server information, entity Meta information and possible entity-body content [5].

2.3 The resource definition framework

The World Wide Web Consortium has developed the Resource Description Framework for the purpose of providing the foundation for metadata interoperability across various resource description communities. One of the main obstacles of the resource description community is the multiplicity of incompatible standards for metadata syntax and schema definition languages. The Resource Description Framework provides a solution to this problem by using the Syntax specification [8] and the Schema specification [3].

The RDF is easily deployable and lightweight due to its design, which is based on Web technologies. It provides interoperability between applications that exchange metadata. It is not only targeted for resource description usage, but also for various other application areas such as electronic commerce, collaborative services, privacy preferences, site-maps and content rating [8].

The main objective of RDF is to support the interoperability of metadata. RDF allows descriptions of WWW resources to be made available in machine understandable form. This allows the semantics of objects to be expressed and exploited [8].

RDF is based on a concrete formal model utilising directed graphs that elude to the semantics of resource description. The basic concept is that a resource is described through a collection of properties called a RDF description. Each of these properties has

a property type and value. Any resource can be described within the RDF. The basic RDF data model is composed of three different object types [8]:

1. Resources

   All things that are described by the RDF expression are called resources. A resource can be an HTML document, XML element, a collection of Web pages or entire Web site or it might not be accessible via Internet at all. But resources always have an URI and they can have optional anchor ids. The extensibility of URI allows the introduction of identifiers for any entity imaginable [8].

2. Properties

   A property is a particular aspect, characteristic, attribute, or relation, which is used to illustrate a resource. A property has always a particular meaning which defines its allowed values, the types of resources it can describe and its relationship with other properties [8].

3. Statements

   A particular resource along with a named property and the value of that property for that resource compose a RDF statement. These three individual parts of a statement are called, respectively, the subject, the predicate, and the object. The object of a statement can be another resource or it can be a literal resource or a simple string or other primitive data type defined by XML. In RDF terms, a literal may have content that is XML mark-up but is not further evaluated by the RDF processor [8].

Figure 6 illustrates the RDF data model. In figure 6 the nodes, which are drawn as ovals represent resources and arcs represent named properties. The nodes that are drawn in the shape of rectangles represent string literals. The direction of the arcs is important. Arcs always starts from the subject, which is the resource, and points to the object of the statement. The diagram illustrated in the figure 6 can also be read as; *"The Person whose name is Markus Sihvonen, e-mail<markus.sihvonenvtt.fi>, is the creator of*

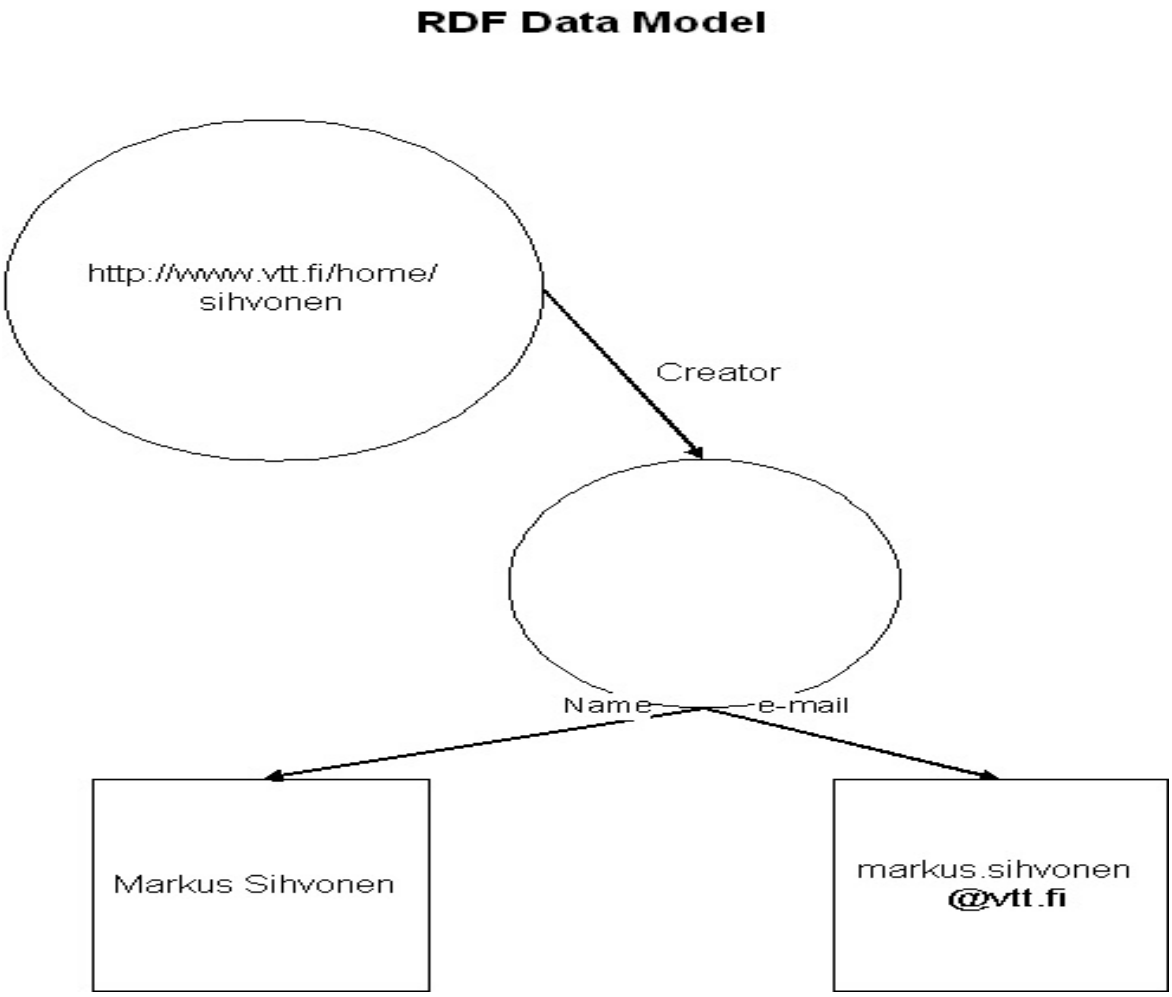


Figure 6. Structured value with identifier.

http://www.vtt.fi/home/sihvonen". The intention of this sentence is to make the value of the creator property, http://www.vtt.fi/home/sihvonen, part of a structured entity. In the RDF this type of entity is represented as another resource. In this example the structured entity does not have a name, which is illustrated as an empty oval in figure 6.

RDF extends the XML model and syntax for describing resources. It utilises the namespace facility of XML. The XML Namespace, which points to a URI, makes it possible for the RDF to uniquely identify a set of properties. The set of properties is called a schema and it can be accessed by the URI, which is identified by the namespace. RDF also inherits all XML syntactic flexibility, which are white space rules, quoting using either single quote (') or double quote ("), character escaping, case sensitivity, and language tags that enable the support of multi-lingual metadata [3].

The namespace for the RDF is declared as:

<RDF xmlns = "http://www.w3.org/1999/02/22-rdf-syntax-ns#">

The URI above addresses the following document:

```
<?xml version="1.0"?>

<RDF
  xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://www.w3.org/TR/WD-rdf-schema#">

<!--
  This is the RDF Schema for the RDF data model as described in the
  Resource Description Framework (RDF) Model and Syntax Specification
  http://www.w3.org/TR/REC-rdf-syntax -->

<s:Class rdf:ID="Statement"
  s:comment="A triple consisting of a predicate, a subject, and an object." />

<s:Class rdf:ID="Property"
  s:comment="A name of a property, defining specific meaning for the property" />

<s:Class rdf:ID="Bag"
  s:comment="An unordered collection" />
```

<s:Class rdf:ID="Seq"

  s:comment="An ordered collection" />

<s:Class rdf:ID="Alt"

  s:comment="A collection of alternatives" />

<Property ID="predicate"

  s:comment="Identifies the property used in a statement when representing the statement in reified form">

  <s:domain rdf:resource="#Statement" />

  <s:range rdf:resource="#Property" />

</Property>

<Property ID="subject"

  s:comment="Identifies the resource that a statement is describing when representing the statement in reified form">

  <s:domain rdf:resource="#Statement" />

</Property>

<Property ID="object"

  s:comment="Identifies the object of a statement when representing the statement in reified form" />

<Property ID="type"

s:comment="Identifies the Class of a resource" />

<Property ID="value"

s:comment="Identifies the principal value (usually a string) of a property when the property value is a structured resource" />

</RDF> [19].

The above namespace declaration also sets the RDF as the default namespace. Other namespaces will be declared as attributes within the RDF tag. The # symbol at the end of the URI is important since it is used to combine the namespace name with the local name to get the full URI of a property type [3].

The RDF specification defines two XML syntaxes for encoding a RDF data model instance. The serialisation syntax expresses the full capabilities of the data model in a very regular fashion. The abbreviated syntax includes additional constructs that provide a more compact form to represent a subset of the data model [3].

The RDF defines three different types of container objects, which are bag, sequence and alternative. The bags are used to declare that a property has multiple values and that there is no significance to the order in which the values are given. The sequence is used to declare that a property has multiple values and that the order of the values is significant. The alternative is used when list of resources or literals represent alternatives for the single value of a property [3].

RDF can be used for making statements about other RDF statements. These are referred as higher-order statements. Before statements about other statements can be expressed, the model from the original statement needs to be constructed first. This model is a new resource to which additional properties can be attached [8].

3. Technology consideration for profile transfer

The Composite Capability / Preference Profile proposal illustrates interoperable capabilities and preferences by encoding a client device's capabilities and references. It is designed for web browsers, application software and peripheral equipment. Although support for devices and application software will require its own specific attributes, the XML/RDF based approach to solve this dilemma can be adequate, since the metadata description scheme can be tailored to its user groups individual requirements [9].

Since a collection of tables is a basic data model for CC/PP, the use of RDF does modelling for a wide range of data structures possible. In the simplest scenario, atomic properties are composed by the RDF statements, which form a table for the CC/PP description. These tables are constructed from manufacturers', possibly non-changeable, default settings, permanent local settings, and temporary changes made by a user, which can also be very dynamic by nature. The manufacturer's settings include both hardware and software unchangeable settings. The changes in permanent local settings contain for example addition to a device's physical memory capacity, and in general the modification is relatively permanent by nature. The temporary changes, for example user preferences, can have very dynamic characteristics. A good example is a user turning sound on and off during a service session that transmits sound. The dynamic control of an end device's profile information is a true challenge for service providers [9].

The profile information must always be associated with the present network session or transaction. A component in a network session or transaction constructs from attributes or user preferences. A component can be a hardware platform or a software platform. The hardware platform is a base element for executing software and the software platform is a host for all the applications to be executed in that environment [9]. The following example illustrates the encoding of a component profile. The data is reproduced from the chapter 1.4.1.

Hardware platform

- Vendor = Nokia
- Model = 9150
- Type = MT
- Screen = Yes
- Screen size = 800*600*24
- Keyboard = Yes
- CPU = PPC
- Bluetooth = Yes
- Memory = 32MB
- Speakers = Yes

Software platform

- Operating system = EPOC1.0
- HTML version = 4.0
- Java Script version = 4.0
- WAP version = 1.0
- WML Script = 1.0
- Sound = ON
- Images = ON

User preferences

- Language = English

Usually, the profile information of one component contains a collection of properties and property values that are common to that particular component. For example in the previous example, the following property values of the hardware platform are always same for that specific mobile terminal type: vendor, model, type, screen, keyboard, CPU and speakers. When these default properties are collected together for a distinct RDF resource, it allows them to be retrieved and cached independently. This technique is not mandatory, but it does improve the performance in relatively slow wireless networks, when proper transmitting strategies are being utilised [9]. The RDF graph is illustrated in figure 7.

The RDF graph consists of nodes, arcs and leafs, which is illustrated by the graph in figure 7. The data for the graph is from the Observation camera scenario in the chapter 1.4.1. The nodes in the graph represent resources, the arcs are properties and the property values are represent as leafs. Figure 7 associates the components with the current network session and the network session serves as a root for the tree structure, which owns the other components. For any given network transaction or network session, only the current CC/PP description has importance. The differences between the default settings and persistent local settings have no valuable meaning. Since it is also possible to obtain CC/PP description from multiple sources, different parts of the capability profile can also be differentially cached. But a proper transmission and decentralisation of profile information strategy must be deployed and the different components must be clearly described in the network transaction or session [9].

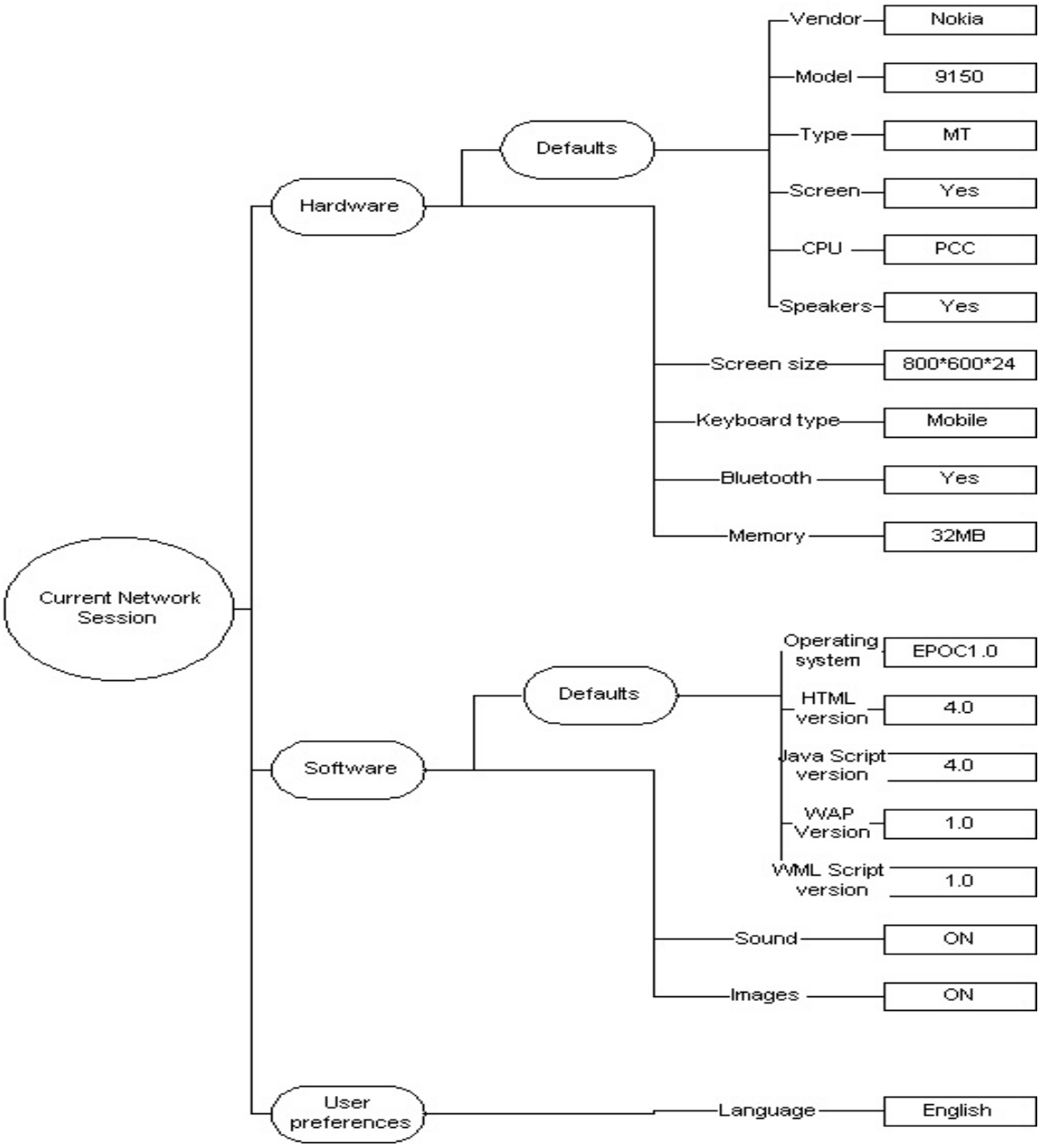


Figure 7. RDF's tree structure associated with a network session.

3.1 Complete transfer of the coded CC/PP description

Consider the alternative where an end user transmits all of its CC/PP description by itself. The following encoding of the Observation camera scenario is based on the XML/RDF syntax [9][4]. This example illustrates information that the Mobile Terminal in the Observation camera scenario may transfer to the home server.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#">
<rdf:Description about="HardwarePlatform">
        <prf:Defaults
        Vendor="Nokia"
        Model="9150"
        Type="MT"
        Screen=”Yes”
        ScreenSize="800x600x24"
        Keyboard="Yes"
        CPU="PPC"
        Memory="16MB"
        Speaker="Yes" />
        <prf:Modifications
        Bluetooth="YES"
        Memory="32MB" />
</rdf:Description>
<rdf:Description about="SoftwarePlatform">
        <prf:Defaults
        OS="EPOC1.0"
        HTMLVersion="4.0"
```

```
                JavaScriptVersion="4.0"

                WAPVersion="1.0"

                WMLScript="1.0" />

                <prf:Modifications

                Sound="ON"

                Images="ON" />

</rdf:Description>

                <rdf:Description about="UserPreferences">

                <prf:Defaults

                Language="English"/>

</rdf:Description>

</rdf:RDF>
```

The third line (xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#") in the example defines the RDF syntax that has been used and the fourth line xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#">) defines the profile vocabulary. Both, the RDF syntax and profile vocabulary are XML name space facilities [4].

The first name space facility that defines the syntax for the negotiation session, http://www.w3.org/1999/02/22-rdf-syntax-ns#, is defined by the World Wide Web consortium. The name space facility defined by the previous URI is more closely illustrated in chapter 2.3, which introduces the resource definition framework.

It is not necessary for a XML namespace URI to point to anything in particular. The exception is when the namespace is used in RDF, then it should point to a schema of the vocabulary [4]. In the example discussed above the URI of the namespace that defines the vocabulary, http://www.w3.org/TR/WD-profile-vocabulary#, does not point to anything. Since the examples used in this study are lacking a vocabulary, the following vocabulary has been designed for use only with the examples discussed in this paper. Whenever there is a reference to ”http://www.w3.org/TR/WD-profile-vocabulary” in this study, a reader should refer to the following vocabulary.

<?xml version=”1.0”?>

<digit>::=’0’|’1’|’2’|’3’|’4’|’5’|’6’|’7’|’8’|’9’

<Default>

Vendor=”NOKIA”|”Siemens”|”Philips”

Model=”9150”|”10150”|”S2003”|”TripleSpark”

Type=”MT”|”MP”|”PDA”

CPU=”PPC”

<Modifications>

Screen=”Yes”|”No”

ScreenSize=”{positive_digit}{positive_digit}+{positive_dig
it}{positive_digit}+{positive_digit}{positive_digit} ”

Keyboard=”Yes”|”No”

Memory=”{positive_digit}{positive_digit}”

Speakers=”Yes”|”No”

OS=”EPOC1.0”

HTMLVersion=”4.0”

JavaScriptVersion=”4.0”

WAPVersion=”1.0”

WMLScript=”1.0”

Language=”English”|”German”|”French”|”Finnish”|”Swedish”

Bluetooth=”Yes”|”No”

Sound=”On”|”Of”

Images=”On”|”Of”

</Modifications>

</Default>[4][2].

When all of the CC/PP description is delivered directly by a client device, some simplifications can be made. If any of the default properties has been changed, there is no need to deliver old values but only the very latest setting. This technique obviously diminishes the transmission load. But still, when a client device itself transmits all the profile information required in a session, a great load is imposed on the network. This may cause a decrease in performance, particularly in relatively slow wireless networks [9].

3.2 Use of remote reference for transferring the CC/PP description

Instead of delivering the complete CC/PP description directly from a client device to a server, a remote reference technique can be used. When the remote reference technique is utilised, a client device transmits URI address of third party to a server and the server retrieves the client device’s default profile information from the location defined by the URI. This is particularly useful when non-changeable default information, such as hardware and software default settings, is being transferred. A separate caching of profile information into functional subsets is enabling this technique. The remote reference technique is most useful when a device that uses a wireless network to communicate with a server and the server is a part of a wire line network that has a connection to the WWW. Naturally, the default information provided by a third party, such as hardware manufacturer, must have been placed into a media that is accessible via the WWW [9].

The following example of an indirect reference also uses the CC/PP description information from the Observation camera scenario. First it provides the user agent’s profile, which also includes URIs of the separately stored hardware and the software manufacturers’ default values [9].

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#">
<rdf:Description about="HardwarePlatform">
        <prf:Defaults>
            <rdf:li resource="http://www.nokia.com/profiles/9150"/>
        </prf:Defaults>
        <prf:Modifications
             Bluetooth=”YES”
            Memory="32MB"/>
</rdf:Description>
<rdf:Description about="SoftwarePlatform"
        <prf:Defaults>
            <rdf:li resource="http://www.symbian.com/profiles/pda"/>
        </prf:Defaults>
        <prf:Modifications
             Sound="ON"
             Images="ON" />
</rdf:Description>
<rdf:Discription about="UserPreferences">
        <prf:Defaults
             Language="English" />
</rdf:Description>
</rdf:RDF>
```

Next there is the profile information that is provided by the hardware manufacturer.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
                <rdf:Description
               Vendor="Nokia"
                Model="9150"
                Type="MT"
                ScreenSize="800x600x24"
                CPU="PPC"
                Keyboard="Yes"
                Memory="16mB"
                Bluetooth="YES"
                Speaker="Yes" />
</rdf:RDF>
```

The software manufacturer provides the last profile information.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
                <rdf:Description
```

```
            OS="EPOC1.0"

            HTMLVersion="4.0"

            JavaScriptVersion="4.0"

            WAPVersion="1.0"

            WMLScript="1.0" />

</rdf:RDF>
```

Because in this example the specific default attributes were grouped together and named by the URI, it is possible to retrieve and cache them separately. This approach leaves the problem of choosing between the default values and user preference values for the application that controls the server logic in the server. The load on wireless networks decreases by using the remote reference technique since the mobile device transfers only the user preferences, permanently modified settings and the URIs of the default settings. The server must request additional information from the mobile device if it did not receive all the required data during the first session [9]. Whether or not there is any acceleration in transferring the CC/PP description of a mobile device to a service provider depends on the efficiency and accessibility of a third party server.

3.3 Dynamic changes in the CC/PP description

The most important information provided to a server by a client device, is its current profile description. At any given moment, the client device and the server must have consistent information on the client device's current profile description [9]. Upgrades, such as increase in memory or a new operating system version, must override the older default information. Also a server must be sensitive to dynamic changes of the client device's current profile, for example the user must be able to turn the sound on and off "on the fly". Naturally dynamic changes in a client device's CC/PP description while a service is being used is a most challenging scenario for a server.

One solution for the maintenance problem is to transmit the entire CC/PP description to the server whenever a change to the current profile occurs. This would then replace the old profile in the server. A solution that is better fitted for slow network is the transfer of the change in the current profile description only. The following example illustrates the situation where the sound is turned off [9].

```
<?xml version="1.0"?>

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:prf="http://www.w3C.org/TR/WD-profile-vocabulary#">

        <Description about="SoftwarePlatform"

        Sound="Off" />

</rdf:RDF>[9]
```

The amount of information, which is required to be sent in the above example, is clearly less than when the full profile information is transmitted. The modification of a client device's present CC/PP description in the server end requires a sophisticated managing application with well-defined rules.

3.4 Discussion

The Resource definition format provides an adequate tool to describe and manage a client device's profile information. Complex multilevel data structures can be described with a simple RDF model. It is easy to define one particular piece of data by a RDF structure and to retrieve it when needed. An addition of new data to a RDF data structure is done by defining a location for that data according to the rules of the structure and then simply inserting it into that location. Because of the versatility of RDF, it is widely used in the Internet community to support the interoperability of metadata and various application areas have evolved to utilise its capabilities. Since RDF is commonly used, it is well known to many software engineers, which advances its deployment.

There are two main strategies for transferring a client device's profile information to a server. One is to transfer the complete profile information from the client device to the server and the other is to use a remote reference to a third party server. The profile information transfer strategy is discussed in chapter 4.

It is paramount importance for a client device to maintain current profile information and to transmit changes to the profile description promptly to a server while a client device is using services provided by the server. Only up-to-date profile information has any meaningful value to a server. The dynamic maintenance requirements must be incorporated into a profile description transfer strategy.

4. The CC/PP protocol strategy

There are three different protocol strategies to consider. The first possibility is to transmit the complete CC/PP description to a server immediately at the start of the session and update the profile information as it changes during the session. The second choice is to transmit as little information as possible to the server at the beginning of the session and then let the server request more information from a client device when it is needed [10]. At each additional profile information request, the client device transmits only the requested data to the server. In this alternative the client device is responsible to transmit changed user profile information promptly to the server. The third possibility is to transmit something between these first two alternatives at the beginning of a session and then transmit the updated user profile information as the changes occur.

The first alternative is good because of its simplicity - all profile information originates from one source. The disadvantage is that it will increase the load of the network. The growth of the network load is not relevant if the used network technology can support the increased load. But most likely, any network architecture will eventually choke on endlessly increasing load and among the first to be overloaded are wireless networks. A clear advantage of the second choice is that it diminishes the load of the network compared to the first alternative. This is a particularly useful strategy for wireless networks. A disadvantage is that the implementation is somewhat more complex than in the first alternative. There is no support for the third strategy, since it is impossible for a client device to know before a server's feedback what information is relevant. In this case a client device either transmits too little information, or irrelevant information. If too little information is being transmitted, the second transmission, where profile information is updated, must occur as in the second transmission strategy. If irrelevant information is being transmitted, unnecessary network load is generated.

The architectures of the different networks vary from each another. The physical sizes of networks and their properties are different. Some networks are accessible only for a limited number of users or in a very limited physical area, which naturally regulates the usage of the network. Some networks are closed without a gate to other networks. The different properties and functionality of networks do affect the decision process of the profile transformation strategy.

Because of the great number of different network types, the strategy for transmitting the profile information of the client device should be flexible. In the situation where a server can access the World Wide Web and the load of the network where the client device is located is heavy, the client device should comply with the transmission strategy two – to transmit as little information as possible. When a network does not have a gate to the World Wide Web, the client device must itself be able to transmit a full profile description to the server. Also when a network has a connection to the World Wide Web, but its load is light, the client device should have an option to transfer full profile information if requested by the server. However the basic assumption is, that during the initial contact a client device transfers the minimum required profile information, which includes only the URIs of the third parties and permanent modification information about hardware and software. Many services may be provided

by the server to a client device based on the knowledge of its default values. Later during the connection, the client must be able to transmit the complete profile information or selected portions of it at the request of the server. According to this profile information transfer strategy, a hardware and software manufacturer's default settings need to be stored in the client device and into a server with public Internet access. The server must be dedicated to the storage of that particular device's default profile information.

According to the chosen CC/PP protocol strategy, a user agent transmits URIs of the default profile information during a first communication session. This approach requires additional intelligence from the server, since it is the server's responsibility to request additional profile information when needed from a user agent. The need for additional information could arise for example when some of the third party CC/PP repositories are not available or user preferences are required to have some particular value before the server can provide a particular service. The discussion of the server logic is beyond the scope of this study and it is left for future studies.

In order to illustrate the chosen CC/PP description strategy consider the Observation camera scenario, in the appendix 1. According to the chosen strategy the mobile terminal initially transmits URIs of the hardware and software manufacturers' default profile information and the following modifications: hardware modifications; Keyboard type = Mobile, Bluetooth=Yes, Memory=32MB, software modifications; Sound=ON, Images=ON. If the server thinks that it does not have fully up-to-date profile information, it must request for more information or transmit an error message. A warning mechanism needs to be incorporated for non-successful events.

The dynamic profile information management must adapt to the chosen CC/PP protocol strategy. The updated profile description must be transmitted to a server immediately but only the changes, not the complete description. The implementation of the profile information management policy requires additional logical decision capabilities from a client device. A clear advantage in increased service quality can be achieved when this policy is utilised in slow networks, but the increased complexity of the client device is a drawback.

5. The CC/PP exchange protocol

The considered exchange protocol should have versatile characteristics. The flexible addressing of a third party URI should be possible and a warning system for transferring error messages and other important information to a client device should be available. In this study the HTTP Extension Framework is a candidate for CC/PP exchange protocol since it is a generic extension mechanism for the HTTP/1.1 [5]. Also in MExE classmark 2, which characterises Java enabled devices with telephony specific extensions, CC/PP descriptions are passed via HTTP. MExE classmark 2 also supports WSP since it is the CC/PP carrier in MExE classmark 1, but HTTP is preferred over WSP in this study since HTTP has additional features over WSP [14].

5.1 Scope and a strength of the extension declaration

The HTTP Extension Framework has two different extension declaration scopes, which are hop-by-hop and end-to-end. Hop-by-hop headers are not stored by caches or forwarded by proxies and they are meaningful only for a single transport-level connection. End-to-end headers are transmitted to the final recipient for a request or a response. The header responses must be stored as part of a cache entry and must be transmitted in any response formed from a cache entry [10]. The following headers type is a hop-by-hop; connection, keep-alive, proxy-authenticate, proxy-authorisation, transfer encoding and upgrade. All other headers are end-to-end type [5]. The extension scope of the HTTP extension declaration should be hop-by-hop if a user agent has the knowledge that the first hop proxy complies with the CC/PP exchange protocol. If the first hop proxy does not comply with the CC/PP exchange protocol or the proxy is not used by the user agent, the extension scope of the HTTP extension declaration should be end-to-end [10].

There are also two types of extension declaration strengths in the HTTP Extension Framework, which are mandatory and optional. If a server does not comply with the CC/PP exchange protocol in a situation where a user agent must obtain an error message the extension declaration should be mandatory. If the user agent must gain a non-tailored content in a situation where a server does not comply with the CC/PP exchange protocol, the strength of the extension declaration should be optional [10].

The choice between different extension declaration scopes and extension declaration strengths depends on the user agent's task. The integrity and the persistence of the extension identifiers must be preserved as long as extension is needed [10].

5.2 HTTP header fields

In this section three different HTTP header fields are discussed, which are the Profile header, the Profile-Diff header and the Profile warning header. The Profile header field contains a list of references that address CC/PP descriptions. The Profile-Diff header

field contains the CC/PP description itself. Both the profile header field and the Profile-Diff header field are request header field types. The Profile-warning header, which is a response-header field, is the tool for delivering warning information [10].

The request-header fields in general allow a client device to transmit additional information about the particular request and information about the client device to the server. The request-header fields act as request modifiers and they have semantics similar to parameters on a programming language method invocation. The general grammar for the request header field is following:

Request-header = accept;

Accept-Chariest;

Accept-Encoding;

Accept-Language;

Authorisation;

Expect;

From;

Host;

If-Match;

If-Modified-Since;

If-None-Matched;

If-Range;

If-Unmodified-Since;

Max-Forwards;

Proxy-Authorisation;

Range;

Referrer;

TE;

User-Agent;

New header fields can have the semantics of the request header field if all parties in a communication recognise them to be request-header fields [5].

A response-header field allows the server to transmit additional information about the response that can not be placed in the status line. These fields provide information about the server and future access to the resource identified by the Request-URI. The response-headers general grammar as follows:

Response-header = Accept-Ranges;

Age;

ETag;

Location;

Proxy-Authenticate;

Retry-After;

Server;

Vary;

WWW-Authenticate;

Also new header fields can follow the semantics of the response-header field if all parties in the communication recognise them to be response-header fields [5].

5.2.1 The Profile header

As previously discussed the Profile header fields' type is a request-header, and it contains a list of references that address CC/PP descriptions. The grammar for the profile header field, which must conform to the grammar of a request-header field, is the following;

Line 1:	Profile	= profile-field-name ":" 1#reference
Line 2:	profile-field-name	= "Profile"
Line 3:	reference	= <"> (absoluteURI | profile-diff-name) <">
Line 4:	profile-diff-name	= profile-diff-number "-" profile-diff-digest
Line 5:	profile-diff-number	= 1#DIGIT
Line 6:	profile-diff-digest	= sp; <MD5message digest encodedby base64>
Line 7:	DIGIT	= <any US-ASCII digit "0".."9"> [10].

The value of the profile header field is a list of references (line 3). Each reference (line 3) in the Profile header field represents the corresponding entity of the CC/PP description. The reference can either be an absolute URI or a profile-diff-name (line 3). An entity of a CC/PP description that is represented by an absolute URI exists outside of the request, and an entity of a CC/PP description that is represented by a profile-diff-name is included into the request (line 3) [10].

The entity that is addressed by an absolute URI, in the profile header field (line 3), must exist in the World Wide Web. It is possible to use multiple absolute URIs, which enable the use of multiple sources for a CC/PP description. According to the chosen protocol strategy the default CC/PP description information that is provided by hardware and software manufacturers, should be accessed by using absolute URIs. The syntax of the absolute URI must conform to RFC2396 [1]. An absolute URI is distinguished from a profile-diff-name (line 4) by the presence of a colon (":") in the profile header field-value [10].

The profile-diff-name (line 3) in the Profile header field addresses a CC/PP description in the corresponding Profile-Diff header within the same request. When the Profile header field includes a profile-diff-name, the corresponding Profile-Diff header must be included within the same request. The Profile-Diff header will be discussed in the next chapter (5.2.2) [10].

The main reason why the profile-diff-name is introduced is to specify the priority of each CC/PP description in the Profile header field-value. The priority is indicated by the order of references (line 3), which can be either absolute URIs or profile-diff-names, in the Profile header field-value. The latest reference in the Profile header field-value has the highest priority. Therefore according the default rule a CC/PP description that is represented by the latest reference can override CC/PP descriptions which are represented by the previous references. The CC/PP schema incorporates an overriding rule for this basic function. When a CC/PP schema is applied, the CC/PP description, which is the latest reference, does not override the previous CC/PP description [10].

The profile-diff-name (line 4) is constructed from the two different partitions: the profile-diff-number partition (line 4) and the profile-diff-digest partition (line 4). The profile-diff-number indicates the corresponding Profile-Diff header. Multiple Profile-Diff headers can be in the same request and they are separated by the profile-diff-number (line 4). The profile-diff-number (line 5) is formed so that it corresponds to the suffix of the corresponding Profile-Diff header field-name (discussed in the chapter 5.2.2) in the same request [10].

Using the MD5 message digest algorithm RFC1321 [11] and the Base64 algorithm RFC2045 [6] to the corresponding Profile-Diff header field-value generates the profile-diff-digest value (line 6). The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit fingerprint of the input. The Base64 algorithm takes as input arbitrary binary data and produces as output printable encoding data of the input. The reason for the existence of the profile-diff-digest (line 6) is the increased efficiency of the cache table to look up gateways, proxies and user agents. When the

server uses some headers to select a representation that is subject to server-driven negotiation, these headers should be included in the Vary header field. In this case only the Profile header is included in the Vary header, instead of including the Profile header and the all-multiple Profile-Diff headers because the profile-diff-digest represents the Profile-Diff header field-value. Now gateways, proxies and user agent look up only the Profile header for the validation of the cache expiration model [10].

The generation method of the profile-diff-name is the following:

1. The MD5 algorithm is applied to the CC/PP description, which is the value of the corresponding Profile-Diff header field.

2. The Base64 algorithm is applied to the output of step 1.

3. Insert the profile-diff-number at the head of the output of step 2 [10].

The two Profile header examples:

Profile:"http://www.aaa.com/hw","http://www.bbb.com/sw"

Profile:"http://www.aaa.com/hw","1-uKhJE/AeeeMzFSejsYshH g= =",

"http://www.bbb.com/sw" [10].

It is possible to incorporate all the references and CC/PP descriptions in one Profile header instead of dividing them into multiple Profile-Diff headers from Profile header. The use of multiple Profile-Diff headers is recommended since;

1. The headers should be small because some implementations of servers and proxies have restrictions on the header length.

2. It is difficult to parse the header field when the profile descriptions and URI are mixed within the same header field-value.

3. The goal of the CC/PP exchange protocol is to gain independence from the conveyed profile format, which means that the mixture of references and profile descriptions is undesirable [10].

5.2.2 The Profile-Diff header

The Profile-Diff header field, which is also a request header field type, contains the actual CC/PP description. The Profile-Diff header field must be used within the Profile header in the same request. When absolute URIs in the Profile header represents all of the profile information, the Profile-Diff header field is not added to the request. The other extreme alternative is that the Profile-Diff header contains all the profile information. In this case, the Profile header includes only the profile-diff-name, which indicates the Profile-Diff header.

The following is the grammar for the Profile-Diff header field:

Line 1:	Profile-Diff	= profile-diff-field-name ":" profile-desc
Line 2:	profile-diff-field-name	= "Profile-Diff-" profile-diff-number
Line 3:	profile-desc	= < the CC/PP description based on XML/RDF text format (any OCTET except CTLs, but including LWS)> [10].

In the Profile-Diff header field, the profile-diff-field-name (line 1 and line 2) is composed of the prefix "Profile-Diff-" and the following profile-diff-number. The profile-diff-number (line 2) indicates the corresponding profile-diff-name (line 4 in the Profile header grammar; chapter 5.2.1) in the Profile header. One request can contain many Profile-Diff headers, and for this reason the profile-diff-number is introduced to indicate the corresponding profile-diff-name in the Profile header. The profile-diff-number corresponds to the prefix of the profile-diff-name in the Profile header field within the same request (line 4 and line 5 in the Profile header grammar; chapter 5.2.1). The profile-diff-number should be increased by 1 when a Profile-Diff header is beeing added by a user agent, a gateway, or a proxy [10].

Two examples of valid profile-diff-field-name:

Profile-Diff-1:

Profile-Diff-10: [10].

The profile-diff-field-name should not have zero as a first digit in the profile-diff-number, because it will cause unnecessary ambiguity in transfer sessions. For example Profile-Diff-02 and Profile-Diff-002 are not allowed in the same request since they can be interpetted to be the same profile-diff-field-name. There never can be similar profile-diff-field-names or profile-diff-numbers included into one request [10].

The format of the profile-desc (line 3) is in accordance with the HTTP/1.1 specification. The HTTP/1.1 header field-values can be folded onto multiple lines if the continuation of a line starts with a space or horizontal tab. All linear white spaces are treated similarly as a space [10].

Below is an example where the Profile header contains the profile-diff-name, which points into the Profile-Diff-header. The data used in the example is from scenario 1.

Profile: "1-P1GRkSjKK50aTWXXndFcSQ=="

Profile-Diff-1: <?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"

xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">

<Bag>

<Description about="HardwarePlatform">

<Defaults>

<Description PRF:Vendor="Nokia"

PRF:Model="9150"

PRF:Type="MT"

PRF:Screen="Yes"

PRF:ScreenSize="800x600x24"

PRF:Keyboard="Yes"

PRF:CPU="PPC"

PRF:Memory="16mB"

PRF:Speaker="Yes" />

</Defaults>

<Modifications>

<Description PRF:Keyboard type="Mobile"

PRF:Bluetooth="YES"

PRF:Memory="32mB" />

</Modifications>

```
 </Description>

<Description about="SoftwarePlatform">
        <Defaults>
                <Description PRF:OS="EPOC1.0"
                                PRF:HTML Version="4.0"
                                PRF:JavaScritpt Version="1.0"
                                PRF:WAP Version="1.0"
                                PRF:WMLScript="1.0"/>
        </Defaults>
        <Modifications>
                <Description PRF:Sound="ON"
                                 PRF:Images="ON"/>
        </Modifications>
</Description>

<Description about="UserPreferences">
         <Defaults>
                <Description PRF:Language="English"/>
        </Defaults>
</Description>

</Bag>
</RDF> [10].
```

In the example above the complete CC/PP description is placed into the Profile-Diff header. According to the chosen CC/PP exchange strategy only the User preferences and the modifications would be represented in the Profile-Diff header at the initial transmission and the default information would be referred to the absolute URI in the Profile header.

5.2.3 The Profile-warning header

The Profile-warning header field type is a response-header field and its function is to carry warning information. When a client device transmits a request along with the Profile header field to a server, the server will request the CC/PP descriptions from the CC/PP repositories by using the absolute URIs in the Profile header field. If any one of the CC/PP repositories is not available, a server can not obtain the fully enumerated CC/PP descriptions, or a server is not able to obtain the most recent CC/PP descriptions a server should respond to the client with the Profile-warning header field [10].

The following is the grammar for the Profile-warning header field:

Line 1: Profile-warning = profile-warning-field-name ":" 1#warning-value

Line 2: profile-warning-field-name= "Profile-Warning"

Line 3: warning-value = warn-code SP warn-target SP warn-text [SP warn-date]

Line 4: warn-code = 3DIGIT

Line 5: warn-target = (absoluteURI | host [":" port])

Line 6: warn-text = quoted-string

Line 7: warn-date = <"> HTTP-date <"> [10].

The warn-code (line 4) has three digits. When the first digit is 1, it indicates the status of the CC/PP description. If the first digit is 2 it indicates the type of the content adaptation applied to the message [10].

The warn-target (line 5) indicates either the absolute URI or the host corresponding to the type of the warn-code. When the first digit is 1, it indicates the absolute URI and if the first digit is 2 it indicates the host [10].

The following list gives the currently defined warn-codes and the recommended warn text:

100 OK

MAY be included if the CC/PP repository replies with first-hand or fresh information. The warn-target indicates the absolute URI, which addresses the CC/PP descriptions in the CC/PP repository [10][5].

101 Used stale profile

MUST be included if the CC/PP repository replies with stale information. Whether the CC/PP description is stale or not is decided in accordance with the HTTP header information with which the CC/PP repository responds (i.e. when the HTTP/1.1 header includes the Warning header field whose warn-code is 110 or 111.). The warn-target indicates the absolute URI, which addresses the CC/PP description in the CC/PP repository [10][5].

102 Not used profile

MUST be included if the CC/PP description could not be obtained (e.g. the CC/PP repository is not available). The warn-target indicates the absolute URI, which addresses the CC/PP description in the CC/PP repository [10][5].

200 Not applied

MUST be included if the server replies with non-tailored content, which is the only representation in the server. The warn-target indicates the host, which addresses the server [10][5].

201 Content selection applied

MUST be included if the server replies with the content, which is selected from one of the representations in the server. The warn-target indicates the host, which addresses the server [10][5].

202 Content generation applied

MUST be included if the server replies with tailored content, which is generated by the server. The warn-target indicates the host, which addresses the server [10][5].

203 Transformation applied

MUST be added by an intermediate proxy if it applies any transformation changing the content coding (as specified in the Content-Encoding header) or media-type (as specified in the Content-Type header) of the response, or the entity-body of the response. The warn-target indicates the host, which addresses the proxy [10][5].

The next examples illustrate two Profile-warning headers:

Profile-Warning: 102 http://www.aaa.com/hw "Not used profile",

202 www.w3.org "Content generation applied"

Profile-Warning: 101 http://www.aaa.com/hw "Used stale profile",

102 http://www.bbb.com/sw "Not used profile",

200 18.23.0.23:80 "Not applied" "Wed, 31 Mar 1999 08:49:37 GMT" [10].

5.3 Discussion

The HTTP 1.1 extension framework contains features required for the efficient communication of a profile description to a server. It contains a mechanism, the request-header, by which a client device can transfer its profile description to a server. A server likewise transmits warning type information to a client device by using the response-header.

The proposed CC/PP protocol strategy does not cause any conflict with the utilisation of HTTP headers instead they support each other. Any number of remote references URIs can be transmitted by one Profile header to a server and the Profile-diff header is used for transferring profile description information from a client device to a server. It is possible to extract just the required piece of the profile description and insert it into a Profile-diff header. Any number of Profile-diff headers may be associated with one Profile header.

6. Conclusions

This study proposes a twofold CC/PP transfer protocol strategy. During the first session a client device transmits only permanently modified default settings and URIs of the locations where the client device's default information is stored. If the server, that provides the requested service, has no access to the third party servers, it must request the default information from the client device. The server can choose to request the default information directly from the client device, even if it can access the third party servers, if it does not impose unnecessary excessive load to the underlying communication network.

The Composite Capability/Preference Profile proposal based on XML and Resource Definition Framework fulfil adequately the requirements for transmitting a client device's profile information. The CC/PP data model is a collection of tables and it enables the use of RDF's numerous data structures. This enables the use of sophisticated data structures to define the capabilities and the profile information of a client device.

The HTTP 1.1 extension framework, which is included into MExE classmark 2, was chosen for the CC/PP exchange protocol. The other possibility would have been WSP, which is included already into MExE classmark 1, but it does not possess all the required features that are necessary for the content capability negotiations between a client device and a server. The MExE classmarks are downward compatible, which indicates that HTTP 1.1 framework will be a suitable extension framework in the future.

At any given moment only the most current profile information of a client device has meaningful value. When a client device receives new capabilities, such as a new application, the information about it must be immediately updated to the profile information folder and when a capability has been removed from a client device the profile information of that particular capability must be removed at the same moment. The same rule is applicable for user profile information.

The proposed transfer methodology works well as illustrated throughout this thesis. The next phase is to build a test environment by means of which the methodology can be evaluated. The test environment will include a PC, which has the role of a client device, and an NT server, which will take on the role of a MExE server. The efficiency of the CC/PP protocol strategy must be tested first. The load imposed on the network and the reliability of the profile transfer are the main results of the first phase.

The design of the server logic is one of the main tasks of the future study. A server must be able to define whether a client device has the required resources to run the requested application or service. Warning information and user instructions must reach the client device. If a client device is missing some vital resource in order to use some application or service, the server must inform the user about it. If the missing resource is software, the server should inform the location to the client device where it can be downloaded or ask permission to download it immediately to the client device.

Next the dynamic profile information maintenance of a client device will be designed. This will test RDF's suitability for describing profile data information. Since RDF has numerous methods to describe data structures, the goal is to define whether or not the tree structure is the best alternative. Various RDF data description models must be constructed for the evaluating purposes.

The common resource vocabulary for the hardware and the software must be designed. This requires collective input from all the participating members in the ITEA-VHE project. At the same time the detailed hardware and software specification should be completed.

References

[1] Berners-Lee, T., Fielding, R., Irvine, U. C. & Masiner, L. RFC 2396: Uniform Resource Identifiers (URI) Generic Syntax Network Working Group, August 1998.

[2] Biggar, Donald R. Bachus-Naur Form.
Available: http://www3.sympatico.ca/dbiggar/BNF.home.html

[3] Brickley, Dan & Guha, R.V. Resource Description Framework (RDF) Schema Specification. World Wide Web Consortium, W3C Proposed Recommendation 03 March 1999.
Aailable: http://www.3w.org/TR/PR-rdf-schema

[4] Elliot, Rusty Harold. 1999. XML Bible. Foster City: CA IDG Books Woldwide ISBN: 0-7645-3236-7

[5] Fielding, R., Gettys, J., Mogul, J. C., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T. HTTP/1.1, Revision 6. HTTP Working Group, November 18 1998.

[6] Freed, N. & Borenstein, N. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Network Working Group, November 1996.

[7] Järvinen, Pertti & Järvinen, Annikki. Tutkimustyön metodeista. Tampere 1993. Tampereen yliopisto, Tietojenkäsittelyopin laitos, Raportti C-1993-2.

[8] Lassila, Ora & Swick, Ralph R. Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium, W3C Recommendation 22 February 1999.
Available: http://www.w3.org/TR/REC-rdf-syntax

[9] Reynolds, Franklin, Hjelm, Johan, Dawkins, Spencer & Singhal, Sandeep.Composite Capability / Preference Profiles (CC/PP): A user side framework for content negotiations. W3C Note 30 of November 1998.
Available: http://www.w3.org/TR/NOTE-CCPP/#CONNEG

[10] Reynolds, Franklin & Nielsen, Henrik Frystyk & staff members of W3C.CC/PP Exchange Protocol based on HTTP Extension Framework. W3C Note 24th of June 1999.
Available: http://www.w3org/TR/NOTE-CCPPexchange

[11] Rivest, R., The MD5 Message-Digest Algorithm. Network Working Group, April 1992.

[12] ETSI TS 22.21 V1.1.0 (1999-01); Technical Specification. Universal Mobile Telecommunications Systems (UMTS). Provision of Services in UMTS – The Virtual Home Environment. Available: http://www.w3.org/

[13] ETSI TS 23.27 V0.1.0 (1999-02); Technical Specification. Universal Mobile Telecommunications Systems (UMTS). The Virtual Home Environment; Open Service Architecture. Available: http://www.w3.org/

[14] ETSI TS 101 438 V7.0.0 (1999-07); Digital cellular telecommunication systems (Phase 2+); Mobile Station Application Execution Environment (MExE); Functional description; Stage 2: (GSM 03.57 version 7.0.0 Release 1998).

[15] 3G TS 23.121 V2.0.0 (1999-04); 3rd Generation Partnership Project;Technical Specification Group Services and System Aspect, Service aspect; Virtual Home Environment. Available: http://www.w3.org/

[16] 3G TS 23.127 V1.1.0 (1999-10); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspect; Virtual Home Environment / Open Service Architecture. Available: http://www.3gpp.org/

[17] 3G TS 23.057 V2.0.0 (1999-12); 3rd Generation Partnership Project; Technical Specification Group Terminals; Mobile Station Application Execution Environment (MExE); Functional description. Available: http://www.3gpp.org/

[18] ITEA
Available: http://www.itea-office.org/

[19] Available: http://www.w3.org/1999/02/22-rdf-syntax-ns#

[20] Available: http://www.mobilemexe.com

Appendix 1

The scenario 1 is illustrated in appendix 1 and the scenario 2 is illustrated in appendix 2.

Scenario: 1

Identification 1/2

Contributor(s): Markus Sihvonen/VTT elektroniikka

Description: Mr. John Smith is visiting his parents in Miami. Just before leaving to Florida he did install a brand new security system to guard his priceless coin collection.

The security systems includes an Observation Camera and a server. Mr. Smiths Home Computer/Server stores the data shoot by the Observation Camera. They both are connected to the wireless (bluetooth) home network. The Observation Camera can rotate 180 degrees vertically and horizontally. A user can create custom movement patterns for the Observation Camera or to keep it is stationary. Its movements can also be remote controlled in real time as well as the images the Observation Camera shoots can be view at any time.

Before going to sleep Mr. Smith decides to look at his coin collection, just to make sure that everything is in order. He boots up his mobile terminal and establishes a connection to his internet service provider. Once the connection has been established he is able to access the Home Desktop/Server (he did establish Home Desktops/Servers internet connection before leaving home) and the Observation Camera. Immediately after the logging into the Home Desktop/Server he is able to gain full control of the Observation Camera.

Scenario: 1

Identification 2/2

Roles:	Mobile Terminal:	• Main controller. • Receives video image.
	Tel. Network:	• Provides the ISDN and UMTS telephone service for ISP.
	Home Desktop / Server:	• The server of the home network • Stores information shoot by the Observation Camera.
	Observation Camera:	• Scan the predefined area or accepts a user initiated control inputs.
	Hardware manufacturer's server:	• The place where the Mobile Terminal's hardware default settings are stored.
	Software manufacturer's server:	• The place where the Mobile Terminal's software default settings are stored.

Scenario 1:

Connection requirements

Link name	Link type	Channel name	From	to	Data type	Bit rate (kbps)	Format	QoS specifics
MT-1	UMTS		MT	PSTN	UMTS	384-2000	Http	
	UMTS		PSTN	MT	UMTS	384-2000	Http	
HD/S-1	ISDN	Modem	HD/S	PSTN	ISDN	64	Http	
	ISDN	Modem	PSTN	HD/S	ISND	64	Http	
HD/S-2	BT	V-In	HD/S	OC	RF-dig	64-384	MPEG-4	A-In synchronized
	BT	V-Out	OC	HD/S	RF-dig	64-384	MPEG-4	A-Out synchronized
SMS-1	ISDN		PSTN	SMS	ISDN	64	Http	
HMS-1	ISDN		PSTN	HMS	ISDN	64	Http	

☞ Additional remarks

Appendix 2

Scenario 2:

Identification

Contributor(s):	Markus Sihvonen/VTT elektroniikka	
Description:	Mr. Smith is visiting VTT electronics in Oulu. While he is conference meeting, his mobile phone receives a fax message. After the meeting he decides to request a possibility to get a paper print out of the fax. It turns out that the VTT indeed has a printer in their facilities that guests can use and also Mr. Smiths fax can be printed out.	
Roles:	Mobile Phone (MP):	• Receives a fax. • Request a printing services.
	VTT-Server:	• Downloads the appropriate software to the MP that enables the printing function • Controls the printing event.
	VTT-Bubble Jet Printer (BJP):	• The device that is used to print the fax.
	VTT LAN:	• VTT electronics' Local Area Network
	Hardware manufacturer's server (HMS):	• The place where the Mobile Phone's hardware default settings are stored.
	Software manufacturer's server (SMS):	• The place where the Mobile Phone's software default settings are stored.

Scenario 2

Connection requirements

Link name	Link type	Channel name	From	to	Data type	Bit rate (kbps)	Format	QoS specifics
MP-1	BT		MP	VTT LAN	Rf-dig	64-384		
BJP-1	CATV		BJP	VTT LAN		10 000		
Server -1	CATV		Server	VTT LAN		10 000		
SMS-1	CATV		SMS	Intern et		10 000		
HMS-1	CATV		HMS	Intern et		10 000		

☞ Additional remarks

Published by

Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and report code of publication

VTT Research Notes 2065
VTT–TIED–2065

Author(s)
Sihvonen, Markus

Title

A user side framework for Composite Capability / Preference Profile negotiation

Abstract

The Mobile Station Application Execution Environment (MExE) is a standard, which is aimed at smart mobile terminals and its purpose is to facilitate intelligent network services. It is a part of the Virtual Home Environment, the mobile extension of the Open Service Architecture. Before a mobile terminal (MT) can download MExE applications, it must transfer its Composite Capability / Preference Profile (CC/PP) description to a MExE server. This master thesis proposes strategies for the transfer of CC/PP descriptions to the MExE servers. The utilised research method is a constructive research method that is based on abstract analysis of written material of the studied subject.

The CC/PP description may originate from multiple sources and there must be a clearly defined strategy for transferring it to the MExE server. A MT must be able to manage dynamically its resources in MExE. This thesis has three research problems: what should be the source of the MT's CC/PP description, what is a feasible CC/PP description transfer strategy for MTs and what are the dynamic resource management requirements of the mobile terminal?

The proposed CC/PP strategy suggests fetching the CC/PP description in fragments from servers in the network and from the MT. This considerably improves the efficiency of CC/PP transfer in mobile networks. Changes to the capabilities of the MT can be transmitted when necessary to the MExE server. MExE satisfies the requirements defined by the thesis and it has potential to become a widely used standard.

The following studies will concentrate on designing decision capabilities for a MExE server and a common resource vocabulary for MTs. The designed server will also support dynamic resource maintenance for MTs. A test environment will be constructed, which allows for a closer evaluation of present and future results.

Keywords
Mobile Station Application Execution Environment (MExE), Virtual Home Environment, Open Service Architecture, mobile terminal, CC/PP description

Activity unit
VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland

ISBN	Project number
951–38–5768–9 (soft back edition) 951–38–5769–7 (URL: http://www.inf.vtt.fi/pdf/)	E9SU00411

Date	Language	Pages	Price
December 2000	English	54 p. + app. 4 p.	B

Name of project	Commissioned by
Middleware for Virtual Home Environment	Mobile Station Application Execution Environment (MExE), Virtual Home Environment, Open Service Architecture, mobile terminal, CC/PP description

Series title and ISSN	Sold by
VTT Tiedotteita – Meddelanden – Research Notes 1235–0605 (soft back edition) 1455–0865 (URL: http://www.inf.vtt.fi/pdf/)	VTT Information Service P.O.Box 2000, FIN-02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374