Realizing Simplicity: Ambient Aware Service Adaptation

Elisabeth-Anna Fischer, Carsten Meyer, Christoph Niedermeier, Robert Seidl
Siemens AG, Corporate Technology
Email: *<firstname>.<lastname>@siemens.com*
Sofia Kapellaki, George N. Prezerakos
National Technical University of Athens, School of Electrical & Computer Engineering
Email: {sofiak, prezerak}@telecom.ntua.gr

***Abstract*—Mobile communication systems beyond 3G will offer ubiquitous connectivity via novel high-bandwidth wireless access technologies, flexible transport mechanisms including QoS guarantees and a large number of exciting new services based on paradigms as ambient intelligence, pervasive computing and context awareness. However, the dark side of this large number of new features is excessive complexity, creating obstacles to effective exploitation and acceptance. The goal of the Simplicity project, supported by the European Union, is to reduce this complexity by: i) providing automatic customization of user access to services and the network; ii) automatically adapting services to terminal characteristics and user preferences; iii) orchestrating network capabilities.**

This paper, after providing a short introduction into the Simplicity project, focuses on the demo scenarios that have been realized within the project. The scenarios illustrate how ambient aware service adaptation can be implemented in our system through the gathering and evaluation of context information based on flexible rules. The demonstrated scenario provides seamless adaptation of services requiring minimal user interaction, thus realizing a Simplicity user experience.

***Index Terms*—service adaptation, ambient awareness, context value chain, network assistance, user preferences, terminal capabilities.**

I. Introduction

As the future mobile communication landscape becomes more and more heterogeneous, offering a large variety of different devices, access technologies, service quality levels and alternative service providers, mobile users will only be able and willing to benefit from these new possibilities if the associated complexity is hidden from them. In particular, processes as configuration of devices, discovery and selection of suitable access technologies, handover between different access networks, service discovery, service negotiation as well as customization and composition of services have to be automated.

Mobile communication systems as GSM or UMTS currently only offer portability of subscriber identities between different mobile terminals. Subscriber information is stored on a chip card (SIM or USIM) that is tied to a particular mobile network and usually restricted to the subscriber identity and some customization information for a limited number of services provided by the operator running the mobile network. Although that concept fulfils its purpose so far, it is lacking generality in the following respects: i) support for alternative wireless access technologies as Wireless LAN, WIMAX etc. is missing; ii) user preferences and customization information regarding arbitrary services, in particular services offered by third party service providers can not be stored on the SIM/USIM card; iii) use of the SIM/USIM card with terminal devices other than mobile phones or devices specifically designed for that purpose is not possible; in particular, it is impossible to use such a card with multiple devices at the same time. As a consequence, neither personalization of services nor seamlessly moving between different devices, heterogeneous networking technologies or composite services contributed by different parties is supported. Therefore, a generalization of this concept that provides a user-friendly solution to the challenges posed by a diverse service and technology environment is required.

The IST Simplicity project's [1] main goal is to simplify the process of using current and future "services" by developing and evaluating a series of tools, techniques and architectures enabling users to customize and use devices and services with minimal effort. The strategy to achieve this goal is based on two major principles: Personalization and Ambient Awareness. Whereas personalization mechanisms facilitate taking into account user preferences and personal usage patterns for configuration of devices, networks and services, ambient awareness mechanisms supply up-to-date information on the state of the user's environment (including information on his device and the state of the infrastructure used as well as his current social context) and thus enable smooth adaptation of services to all aspects characterizing the user's current situation.

In section II the basic ideas hiding behind the Simplicity concept are presented, while in section III a high level description of the architecture that supports and implements those concepts is given. Finally in section IV the mechanism for the ambient aware adaptation of services is provided, as well as the application of this mechanism in the framework of IST Simplicity project for specific implemented scenarios. Conclusions of our work are summarized in section V.

II. THE PHILOSOPHY BEHIND SIMPLICITY

As already mentioned the two major principles of Simplicity are Personalization and Ambient Awareness.

The personalization concept used in the project is based on a “user profile”. Each user will be able to create and maintain one or more personal profiles, providing access to different services, perhaps using different classes of terminals [2], [3]. Creating and maintaining user profiles will involve also the automatic processing of behavioural information. More refined policies on how to handle specific types of personal information will be part of the user profiles and will be controlled by the user. Control of personal data, security of information, and user privacy are key issues for the Simplicity approach.

The personalized user profile will allow: i) transparent customization and configuration of terminals/devices and services; ii) uniform mechanisms for recognizing, authenticating, locating and charging the user; iii) policy-controlled selection of network interfaces / access networks and application services.

In addition, the user profile will support the second important feature of the Simplicity concept: Ambient aware service customization and adaptation. The profile can hold rules/policies that influence or control ambient awareness mechanisms as automatic selection of devices, networks and services appropriate to specific locations (e.g., the home, buildings, public spaces) and situations (e.g. urgent business issues; user is in a private communication context where special cost restrictions apply), or automatic adaptation of network connections and application services to the capabilities of specific devices, network infrastructures and the user’s situation. The latter may be characterized by user preferences specified in the profile as well as arbitrary types of context information provided by mechanisms residing on the terminal (e.g. a date book application providing appointment data, a microphone providing information on the noise level, arbitrary terminal-based sensors) or in the network (e.g. context brokers offering information collected from network management entities, network-based sensors or other terminal devices).

In order to support and provide all the principles and features described so far, a distributed brokerage architecture [4] consisting of different types of Simplicity entities has been designed based on user scenarios and business models. In the following section a more analytical description of the Simplicity architecture is provided.

III. HIGH-LEVEL ARCHITECTURE

SIMPLICITY is based on a distributed architecture where software and hardware entities are organized into logical groups called SIMPLICITY Brokers (SBs). SBs can be instantiated on user’s terminals as well as in the network and are of different kinds offering different services and data. The core of the SB is an entity called Mediator to which all other local entities are attached. The Mediator implements an event based communication scheme between Simplicity entities. This communication scheme is extended across brokers by the Simplicity Broker Communication (SBC) mechanism, so that it is possible for entities residing at different SBs to exchange events. The SBC reuses existing protocols and communication paradigms (e.g. current implementation uses SOAP and SIP) as much as possible. This distributed nature allows for the realization of different conceptual views of the overall architecture. With respect to adaptation and personalization, SIMPLICITY can be viewed in terms of the following logical hierarchy (see Figure 1):

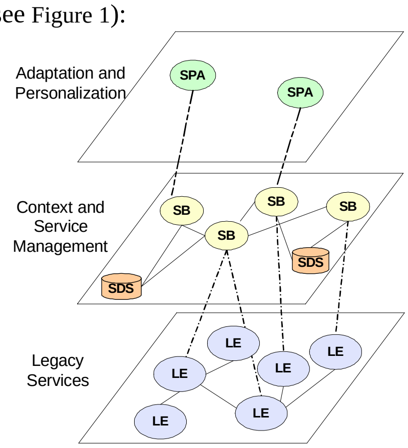

Figure 1: Overall picture of Simplicity architecture

The top layer contains entities which provide functionality with regard to adaptation and personalization of services. Those entities usually directly interact with the user. The middle layer hosts all entities that provide the Simplicity Network. Functionality provided here includes the gathering of sensor information and enriching it into context as well as service discovery and announcement. The bottom layer contains all entities which are not Simplicity-aware (Legacy Entities – LEs).

The Simplicity Personal Assistant (SPA) is the entity that performs high-level personalization and service adaptation tasks as well as context evaluation, learning, rule generation, and service subscription/ orchestration/ invocation. SPAs interact with users via a convenient (and adaptive?) user interfaces. SPAs are meant to provide as much support as possible to the user. Their look, behaviour and actions are strongly adapted to user preferences and needs. SPAs operate on the very “top layer”, interacting with the user and also with other elements of the Simplicity system. They act autonomously whenever they can, requiring only minimal input from the user. The SPA is part of the SB at the terminal side.

Simplicity Brokers (SBs), residing in the middle layer, are responsible for network-based profile and capability management as well as ambient awareness mechanisms like collection and sharing of sensor/context information, context aggregation and refinement etc. These SBs enable easy and flexible integration of different autonomous functionalities such as (but not limited to): collection and combination of information from other entities, provision of the resulting information (context, preferences, policies) to other entities, management of network-related functionalities (e.g. advanced mobility management), and provisioning of services.

Simplicity Devices (SDs) also reside in this layer. The SD constitutes a personalization device that provides a simple and uniform mechanism for customization of terminal devices, network access and services. The user profile will be either stored in the SD or alternatively, storage on the SD might be limited to a "pointer", making it possible to download the profile from the network. The virtualization of the SD performed by abstracting from a physical device is an important step. The resulting generalized concept includes provision of profile data by network entities as well as software agents. In particular the latter facilitate virtual cloning of a SD in order to allow simultaneous personalization and control of multiple terminal devices. Interesting applications of this feature include migration of services from one device to another or the split-device approach that involves two or more devices for using particular services. The SD interacts with other entities as part of the respective SB. The way in which this interaction is handled depends on the implementation of the SD. A "powerful" SD may even physically host a SB; a "dumb" SD is attached to the SB by means of software adaptation.

The bottom layer is dedicated to Legacy Entities (LEs). LEs include all entities that are not Simplicity-enabled/configurable, but are accessible and configurable through a non-Simplicity interface via an adaptor. This adaptor enables the seamless integration of LEs into Simplicity and enhances LE functions with Simplicity-specific features. Examples of LEs are legacy services, sensors etc. that interact with SIMPLICITY without being directly attached to a SB.

IV. Aggregation & Exploitation of Adaptation Data

In order to illustrate how the described architecture serves the Personalization and Ambient Awareness vision presented in the previous sections, a number of demo scenarios and their underlying mechanisms are described in detail in the following.

The user scenario that provides the basis for the demo scenario analyzed in this section is the one of the Mobile Worker [5], [6]. The "Mobile Worker" represents the contemporary user who by utilizing different terminals, applications, services and networking technologies fulfils his tasks, while moving from one place to another. The difference in our scenario is that our "Mobile Worker", doesn't need to handle by himself the complexity caused by his environment's diversity. Everything is handled by the SPA residing in his Simplicity enabled terminals, by his SD and by the Simplicity Brokerage Framework.

The basic idea of the following scenarios is the ambient aware adaptation of services. This means that the services adapt to the user's needs and his situation, as well as to device capabilities and available resources.

In order to be able to adapt services and contents, the Simplicity infrastructure needs information about different users. For this purpose the concept of the "context value chain" is employed [7]. It describes how through a process of sensing of raw data, followed by a series of refinement and aggregation steps adaptation information for services is generated.

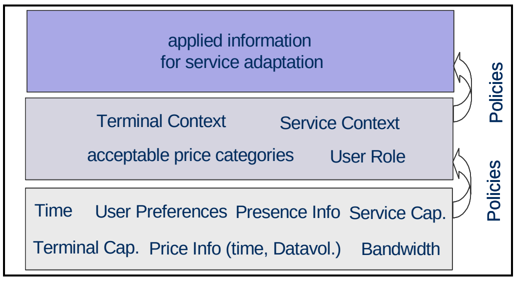

Figure 2: Generation of adaptation information

The first step is made up of the sensing of raw data through sensors which by themselves do not contain any high-level information. This data are then subjected to a sequence of refinement steps. Among other things, refinement comprises the transformation between different formats of representation (e.g., from GPS coordinates to street names and numbers), the extension of context information with attributes (e.g., to express its accuracy), or the combination of context information to derive another one (e.g., determining the user role). The different kinds of refined information are then aggregated in order to be available for evaluation. Building upon this aggregated information, contextualized information can be derived. It must be stressed that aggregation of data may comprise several independent sequences of sensing, refinement and aggregation, one for each kind of context information to be considered. These sequences might be executed in a prescribed order or in parallel, either entirely independent from each other or with need for synchronization. Parts of the raw data are provided by the user and the device itself. This is done when registering at the system. The software automatically transmits the personal preferences as well as the terminal capabilities to the context subsystem, which registers for changes on this data. But raw data is not only provided by the device and the user, but also by the infrastructure itself, whereas tariff information, available bandwidth, time and location information about the user is provided by the network. As this data changes quite frequently, the context subsystem subscribes for this information. Whenever this data changes the context subsystem is informed. In order not to send an update every second, the context subsystem defines a threshold which defines the margin when to send updated raw data.

As raw data can not be directly used to adapt the service, the context subsystem has to "calculate" some kind of data that contains more information about how the service can be ideally used by the user. The context management therefore takes the raw data and refines it to "context data". This kind of data is of higher value than the raw data, since it already represents data that can not be provided by a single source. The context data is then again refined and aggregation data is calculated out of it. Figure 2 contains examples for raw data, context data and adaptation data.

Generation and processing of context information is controlled by policy rules in our system. These policy rules specify how raw data are transformed into context data which in turn are transformed into adaptation data. Policy rules for transforming raw data into context data are general, not application-specific. Policy rules describing the transformation of context data into adaptation data are application-specific, since adaptation data have to support the application. Adaptation data is the most complex information, in comparison to context data and raw data, because their generation involves complex data and policy rules. If the relevant context data is changed, the adaptation information is re-processed, so that it always reflects up-to-date information. Each application that has subscribed for adaptation data can react automatically to a change in the environment that enables the general vision of Ambient Awareness.

A. Demonstrator Setup

All demonstration scenarios described in this paper are aimed to realize an ambient aware multi-party chat. Three different users participate in a multi media chat session. All devices used by the participants are different. One participant in the chat session sends a multimedia message (e.g., a picture) to the other two participants (Push-to-Show). The session setup occurs via a "Push over Cellular" (PoC) server. The multimedia content that is transferred is adapted to the context of the respective receiving terminals. This context is derived from various kinds of information: user preferences, network capabilities, the service pricing model, the user's current role, etc. The provisioning of context information is performed by the SBs: they are in charge of collecting, processing and exchanging context information within network domains, and across network domain borders. A media gateway that is controlled by the PoC Server performs adaptation of the multimedia content.

The demonstrator system consists of three terminals, two of which are within the same network domain. Each network domain also hosts a SB, which is in charge of collecting, processing and distributing profile and capability data, and also contextual information. One domain also hosts the Media Gateway (MG) and a PoC server. The PoC server orchestrates the chat session. The MG is in charge of transforming incoming and outgoing multimedia content.

Figure 3 shows the setup of the Simplicity demonstrator and the messages that have to be exchanged when sending an image from one user to another. The blue arrows show the initial data exchange that takes place when registering at the Simplicity environment. Step (1) is the invite message that is sent to the PoC Server, which relays it to the chat partners and requests data about the chat partners. The partners send an accept message (3) back to the PoC Server. Step (2) is the request for raw/context data which is send from the PoC server to the Simplicity broker of the domain. As the chat partners are not part of domain A, the broker requests the context data of the other users in the other domain. In order not to have the context data calculated twice, the broker directly requests context data, and no raw data. It also subscribes for being informed about changes in the context data. The collected context information is transformed into adaptation information, and transmitted to the PoC server (4). In step (5), the PoC instructs the MG about the necessary transformations. The MG checks whether the requested transformations are actually supported, and responds with an acknowledgement in step (6).

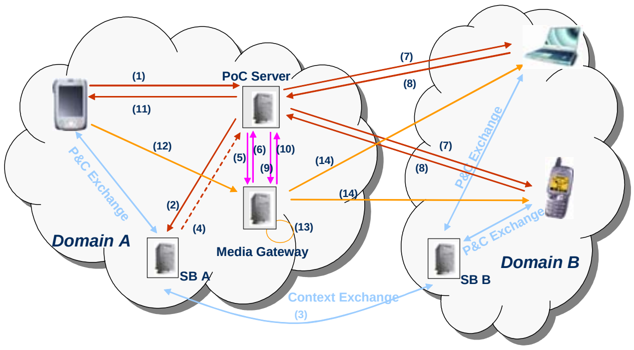

Figure 3: Demonstrator setup

The session initiation now proceeds, with an invitation to the other clients. They are invited by the PoC server, and they respond to the invitation (possibly with limiting modifications, regarding the kind of multimedia data to be used) in steps (7) and (8). From the point of view of the PoC server, all necessary information about all participating parties has been gathered. In steps (9) and (10), the PoC server once again informs the MG about the requested transformations, this time based on the session preferences of all parties. In step (11), the PoC server informs the initiating client that the session has been successfully initiated, and that multimedia messaging may commence. In step (12), the sending client sends the intended multimedia content (for instance, an image) to the MG. The actual transformation takes place in step (13). Afterwards, the transformed multimedia content is sent to the intended recipients (14). A session update may occur at any time if the context of a user changes. This may result directly in a change of the adaptation information.

Triggers for this update may come from different sources. If users manually update their preferences, a direct change in the context information results and this must be acted upon immediately. Another possible trigger for this update is a change in the tariff for contractual reasons (e.g., multimedia messaging is cheaper between 18:00 and 06:00). Both triggers may require a change in the format of the multimedia information on the receivers' side. For instance, the resolution of an image or a video may change, in response to a change in context. The following two subsections describe two different adaptation scenarios. In the first scenario, data adaptation is performed in the network whereas in the second scenario data adaptation on the sending terminal is performed as well.

B. Data Adaptation within the Network

In this scenario the user is taking part in a multimedia session with partners. The session is already set up and the multimedia chat is established. After some time the user changes the cost limit for the service and saves it in his preferences. The modified user preferences are instantly updated by the responsible SB, which in turn updates all context information that depends on the respective user preferences. The PoC server is then informed of the context change, and instructs the MG to change its adaptation behaviour accordingly. All multimedia content that is sent from now on will be adapted according to the changed preferences.

The adaptation of the media data can not only be triggered by user preferences, but also by changes of sensor data or other context information. The following scenario describes the adaptation of received multimedia content due to a change of the location of the user's terminal.

As in the scenario above the user is in a multi media session when he has a doctor's appointment. He enters the doctor's surgery and sits in the waiting area. The location information is updated, and results in a change in the user context. According to the specified context evaluation rules, the user does not want to receive audio information in this particular situation ("at the doctor's"). The SB instructs the MG, via the PoC, to filter out all audio information from messages addressed to this user, before sending them on. This reduces cost, and puts less strain on the air interface than a receiver-based filtering mechanism. When the user leaves the doctors place, the location information is updated and audio information is sent again.

The above described scenarios show the adaptation of data within the network, resulting in a reduction of network traffic. But the most efficient way of optimizing traffic would be to transform the data before it is sent. A scenario that is based on sender side adaptation is being discussed in the next section.

C. Data Adaptation on the terminal

This showcase describes the sender-side adaptation of multimedia content due to group context. In this scenario the session is not established yet. User A sends an invite to user B and user C. User A wants to establish a multi media session. The MG therefore relays the invite to B and C which then transmit their maximum display resolution. In this example user B has a maximum resolution of 240 x 320 pixel (QVGA), user C 132 x 176 (QCIF) and user A 480 x 640 (VGA). In order to optimize the traffic sent to the network, the SB automatically calculates the maximum resolution that has to be sent, which is in this example QVGA (max. (QVGA, QCIF)). The SB instructs the application to send all images in the maximum resolution of QVGA, and not VGA, as it would have been done without Simplicity.

This approach provides the possibility to perform an ambient aware optimization of bandwidth consumption not only within the network, but also over the air both for the sender and receiver sides.

V. CONCLUSION

The Simplicity project addresses crucial issues for future systems beyond 3G: It proposes a solution to handle the increasing complexity of systems, services and technologies. By implementing a demonstrator scenario that is built upon the flexible distributed architecture designed in this project, we were able to show how a user can benefit from the Simplicity approach. We demonstrated how ambient aware service adaptation can be implemented in our system through the realization of context value chain that gathers all necessary data and evaluates it based on flexible policy rules. Among the benefits for the user are the minimal interaction and seamless adaptation of services.

Future aspects to be researched in this project may include security issues involved when realizing the distributed broker architecture across network and operator domain borders. Data access and transfer rules have to be implemented and enforced in order to guarantee the user maximum privacy while still being able to gather information for necessary evaluations.

ACKNOWLEDGEMENT

This work has been performed in the framework of the IST project IST-2004-507558 Simplicity, which is partly funded by the European Union.

REFERENCES

[1] IST Simplicity project: http://www.ist-simplicity.org

[2] N. Blefari Melazzi et al. "The Simplicity Project: Managing Complexity in a Diverse ICT World", in Ambient Intelligence, IOS Press (in press)

[3] N. Blefari-Melazzi, G. Bianchi, G. Ceneri, G. Cortese, S. Kapellaki, K. Kawamura, C. Noda, S. Salsano, I. S. Venieris: "The Simplicity project: easing the burden of using complex and heterogeneous ICT devices and services. Part I: Overall Architecture", IST Mobile&Wireless Communications Summit 2004, June 27-30 2004, Lyon, France.

[4] N. Blefari Melazzi, G. Ceneri, G. Cortese, N. Davies, N. Dellas, A. Friday, J. Hamard, E. Koutsoloukas, C. Niedermeier, C. Noda, J. Papanis, C. Petrioli, E. Rukzio, O. Storz, J. Urban: "The Simplicity Project: easing the burden of using complex and heterogeneous ICT devices and services. Part II: State of the Art of Related Technologies", Mobile&Wireless Communications Summit 2004, June 27-30 2004, Lyon, France.

[5] R. Seidl, F. Berger, S. Kapellaki, T. Frantti, E. Rukzio, J. Hamard, N. Blefari Melazzi: "The Simplicity Project: Personalized and simplified communication spaces for mobile users", Mobile&Wireless Communications Summit 2004, June 27-30 2004, Lyon, France.

[6] D2101 "Use cases, requirements and business models", www.ist-simplicity.org

[7] H. Hegering, A, Küpper, C. Linnhoff-Popien, H. Reiser: „Management Challenges of Context-Aware Services in Ubiquitous Environments", In Self–Managing Distributed Systems; 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2003, Heidelberg, Germany, October 2003, Proceedings, (LNCS 2867), 246–259, Springer, Heidelberg, Germany, Oktober, 2003.