# The Simplicity project: easing the burden of using complex and heterogeneous ICT devices and services.
## Part I: Overall Architecture

N. Blefari Melazzi[1], G. Bianchi[1], G. Ceneri[2], G. Cortese[3], S. Kapellaki[4], K. Kawamura[5], C. Noda[5], S. Salsano[1], I. S. Venieris[4]

[1] DIE, Università di Roma "Tor Vergata", {giuseppe.bianchi, blefari, stefano.salsano}@uniroma2.it

[2] Radiolabs, Consorzio Università Industria – Laboratori di Radiocomunicazioni, Rome, gianni.ceneri@radiolabs.it

[3] Telecom Italia Learning Services, g.cortese@computer.org

[4] National Technical University of Athens, School of Electrical and Computer Engineering, Intelligent Communications & Broadband Networks Laboratory
sofiak@telecom.ntua.gr, ivenieri@cc.ece.ntua.gr

[5] DoCoMo Communications Laboratories Europe, {nick, noda}@docomolab-euro.com

**ABSTRACT**

As technology develops, people are using an ever broader and heterogeneous range of ICT (Information and Communication Technology) devices and network-based services. New areas of research, such as pervasive computing, will further increase the diversity of the devices and services with which users have to deal. The result is an enormous burden of complexity on the shoulders of users, service providers and network operators. Excessive complexity, in turn, creates obstacles to effective exploitation and acceptance of beyond 3G systems such as ambient intelligence, context-aware services and novel access technologies. The goal of the Simplicity project 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.

Simplicity stands for Secure, Internet-able, Mobile Platforms LeadIng CItizens Towards simplicity. However, the project acronym intends to convey also the very aim of the project: develop and evaluate a series of tools, techniques and architectures enabling users to customize and use devices and services with minimal effort.

### I. INTRODUCTION

The Simplicity project [1] started from a vision: that of a user surrounded by different terminals and devices, which, in principle, allow her/him to access several "services". Here, we give to services a very general meaning: from communications to computing facilities, from home/building/public-space functionality to security related tasks. Services may be as simple as remote control of an entertainment device (e.g., a television) via a wireless link, or access control to a building. Conversely, services may be very complex, and may require location awareness, QoS support, message exchanges with network databases, structured interaction with remote networking devices (e.g., media gateways), etc. The emergence of new research areas, such as pervasive computing, will further increase the diversity of the devices and services with which users have to deal.

To exploit all such "services", users must frequently use different devices, configure each of them by using different procedures, must be recognized and authenticated in different ways, must be charged with different means and must use heterogeneous access technologies and protocols. This places an enormous burden of complexity on the shoulders of users (and, often, implies the physical burden of carrying different devices). Such complexity is likely to limit the effective exploitation of the wide range of heterogeneous access technologies, virtual reality, ambient intelligence and context-aware solutions currently under study and development. In fact, it is clear that the average citizen will have difficulties in understanding all these concepts, let alone using them effectively. Not only will this slow down the deployment of new services, it will also help contribute to the digital divide, making it difficult for non-technical users to benefit from new developments. In addition, network operators have to deal with the complexity of a multi-access networking environment.

The aim of this project is to simplify the process of using actual and future "services". In more details, the project goal is to design and deploy a brokerage level able to i) personalize service fruition as a function of user preferences and needs, ii) allow seamless portability of services and running applications/sessions through heterogeneous terminals and devices, and iii) smoothly adapt service fruition to the underlying networking and service support technologies and capabilities.

This paper describes the basic challenges that drove the project proposal and sketches the preliminary ideas developed in the first months of the project lifetime.

The main objectives of the Simplicity project are:

- Describe user scenarios and business models for the Simplicity approach;
- Explore new brokerage mechanisms and policies in a multi-access networking environment.

- Design a universal multi-application Simplicity Device to provide users with a simple and uniform mechanism for customizing services and terminals.
- Validate feasibility and benefits of the Simplicity approach within a test-bed.

## II. DRIVING CONCEPTS

An important lesson learned from 2G wireless systems, e.g., GSM, is the portability of the user identity through different mobile phones. We argue that the generalization of portability, from the basic user identity, to a rich scenario encompassing complex and distributed applications and services over widely heterogeneous networking technologies and devices, may represent the right way towards a simple, user friendly, fruition of the surrounding technologies.

The personalization concept is primarily based on a user profile. In our view, each user should be provided with a personalized profile to be used for different services, eventually using different classes of terminals. In order to create and maintain a personalized user profile, behavioural information will be automatically processed. However, the user should have at least the possibility to control the information gathered about her/him by either initially cancelling automatic storage or deleting specific stored information. More refined policies on how to handle specific types of personal information will be part of the user profile and can be controlled accordingly by the user. Full control of personal data, security of information, and user privacy are key issues of the Simplicity approach and therefore a major concern of the project.

The personalized user profile should allow: i) an automatic, transparent customization and configuration of terminals/devices and services; ii) a uniform way for the user to be recognized, authenticated, located and charged; iii) a policy-controlled selection of network interfaces and applications services. Thanks to this profile, users could also enjoy the automatic selection of services appropriate to specific locations (e.g., the home, buildings, public spaces), the automatic triggering of home/building/public-space functionality, the automatic adaptation of the information delivered, and the easy exploitation of different telecommunications paradigms and services.

Depending on user's characteristics, preferences and abilities, the personal profile could take the form of e.g.,: i) a standard profile defined by a Service Provider; ii) a pre-defined template whose parameters can be configured by the user; iii) an open profile designed by the user using a high-level description language.

Such a user profile is stored (or partially stored and network-downloadable) in a so-called Simplicity Device (SD). Though it seems natural (from our own everyday experience of 2G systems) to think to the SD as a physical device (e.g., enhanced SIM card, Java card, Java ring, USB pen, etc.) we remark that, in greater generality, the SD may be a network location or a software agent. If the SD is a physical device, users could personalize terminals and services by the simple act of plugging the SD in the chosen terminal (see Figure 1).

One of the main novelties of the SD is that it is not bounded to a single networking environment, or to a class of user terminals. We would like to be able to plug the SD into different terminals, providing the necessary additional information and a user-tailored configuration, able also to adapt itself both to the user terminal and to the ambient environment. This means, for instance, that:

- Different users plugging their SDs in the same laptop should see different working environments, file systems, software tools, connection services, etc;
- A given user who plugs the SD in different terminals should see the same representation (adapted to the terminal characteristics) of his personalized working environment;
- A user should be able to suspend and later on resume running applications/sessions by simply unplugging/plugging the SD from/into a terminal, meanwhile finding the application/session automatically adapted to the new context.


**Figure 1: Reference scenario**

## III. SYSTEM ARCHITECTURE

The Simplicity Device will provide a uniform model for the personalization of terminals, network access and services by storing personalization rules for the individual user. However, the SD requires to be deeply integrated in a system framework able to decouple user needs and service deployment and use from the underlying terminal devices, networking technologies and service support platforms. The goal of the Simplicity system is to manage all the complexity, while limiting the end-user interaction to special decisions. To describe the system architecture, we use a horizontal (layered) view and a vertical (user/terminal/network roles) view, as follows.

### *A. Horizontal view*

A first step in the architecture design is to define a logical architecture, showing layers and components for service provisioning, rather than defining the location of explicit software components in specific physical nodes (such as devices and networks). The logical architecture of the Simplicity System is defined in terms of a layered approach, to increase re-usability of middleware components. As shown in Figure 2, and following [2], this middleware consists of the following layers: User Support Layer, Service Support Layer and Network Support Layer.

The **User Support Layer** supports autonomous and proactive agent functions, providing personalization, adaptation and coordination components, which are lacking in traditional middleware. It enables simplifying user interactions with the system and providing user-centric services, by analyzing user contexts and user preferences.

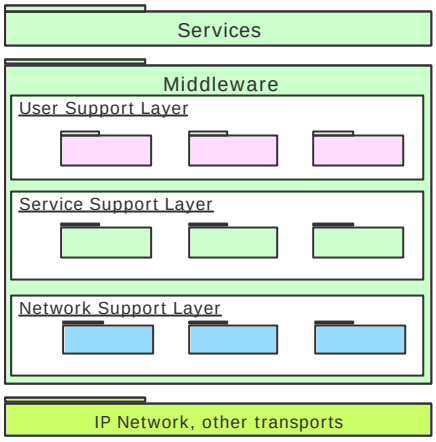


**Figure 2: Logical architecture: layered view**

The **Service Support Layer** contains traditional middleware components, such as advertisement/discovery, profiling and security functions. A dynamic service delivery pattern is introduced to enable a powerful interaction by using discovery, advertisement, authentication, and authorization functions. It is used to negotiate the conditions of service delivery according to users needs.

The **Network Support Layer** provides components for network communications control in heterogeneous networks such as mobility management and QoS management.

*B. Vertical view*

The separation of the operations to be performed at the user side from those to be performed at the network side is crucial. Since a number of possible solutions may be found, a trade-off comparative analysis will be carried out to understand which solution best suits our needs. Figure 3 depicts the basic components we expect for the Simplicity System, and a first proposal for the separation of functionality among them. The proposed system encompasses three main components: the Simplicity Device, the Terminal Broker and the Network Broker.

The role of the **Simplicity Device** (SD) is to store user's profiles, preferences and policies. It also stores and allows the enforcement of user-personalized mechanisms to exploit service fruition, to drive automatic adaptation to terminal capabilities, and to facilitate service adaptation to various network technologies and related capabilities.

The **Terminal Broker** is the entity that manages the interaction between the information stored in the SD and the terminal device as well as the network technology and capabilities. The Terminal Broker enables the SD to perform actions like terminal capability discovery, adaptation to networking capabilities and to the ambient, service discovery and usage. The Terminal Broker caters also for the user interaction with the Simplicity system.

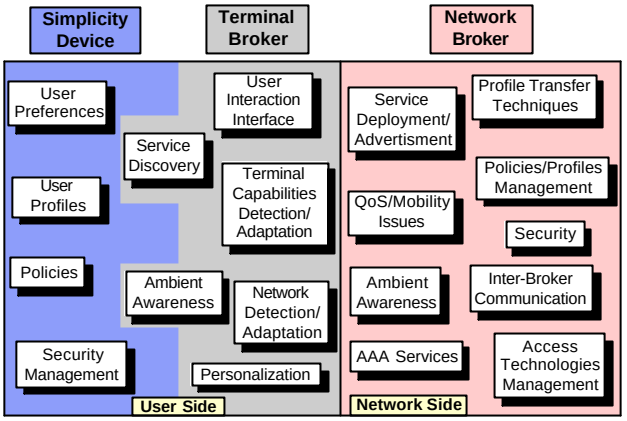


**Figure 3: Physical architecture: component view**

The **Network Broker** has the goal to provide support for service description, advertisement and discovery. Moreover, it orchestrates service operation among distributed networked objects, taking into account issues related to the simultaneous access of several users to the same resources, services, and locations. It also share/allocate the available resources, and manage value-added networking functionality such as service level differentiation and quality of service, location-context awareness, and mobility support.

## IV. SYSTEM FUNCTIONALITY

For simplicity of presentation, the following description is organized according to the vertical/component view (Simplicity Device, Terminal Broker, and Network Broker), though we recall that the outlined functionality will take significant advantage, during their development, from the horizontal (layered) view discussed in Section III.

*A. Simplicity Device functionality*

The key role of the Simplicity Device is to store in a secure and safe way user's profiles, preferences and policies to allow dynamic and automatic discovery and registration of terminal and network capabilities. The SD also facilitates service adaptation to various network technologies and related capabilities. The SD is an important entity of the architecture, since the user cannot access the Simplicity System and exploit its benefits without having one, even if a Terminal Broker resides in his terminal and a Network Broker is available. The SD is the enabling factor and it can be seen as the passport of each user for her/his entrance to the Simplicity world.

A basic goal of the Simplicity project is to give the SD a "universal" potential so that it can be used over a wide range of consumer terminals (from legacy to emerging ones), and it can interact with different environments. The environment can consist of one or several devices. A device can be the device to which the SD is attached to or embedded (e.g., SD in a mobile phone, laptop, PDA, desktop, watch), or an external one, like servers or other home and office appliances. Devices can be sensors

(provides information), actuators (can be triggered by the SD to perform actions) or processing devices with input and output. The SD can be integrated or plugged into other devices in the environment (specifically, mobile devices) or interact with them through short-range wireless networking e.g., BlueTooth. Such integration is possible both for software and hardware implementations of the SD. For the above reason, user profiles, device capabilities, preferences, policies, etc. should be defined and recorded in the SD by means of a high-level description language, such as XML. In this way, it is possible to provide an abstract functionality layer, globally applicable.

The SD requires memory for storage of information. Such memory can be located internally, on the SD, or externally, on the devices in the environment. To this purpose, the SD can store a set of pointers to network locations (e.g., Profile Repositories, see section IV.C). Appropriate entities of the Terminal and/or Network Broker could retrieve and process these pointers, e.g., to download essential software that can not be stored in the SD (e.g., for storage limitations). Such a download should take place via the Network Broker, from specified locations to the terminal currently personalized with the SD.

*B. Terminal Broker functionality*

The user terminal/device is supplied with a so-called Terminal Broker (TB). The TB is the entity that "interfaces" the user to the network.

At this preliminary stage of the project, it is not yet clear how functional roles will be separated between the SD and the TB. In fact, this separation depends on a number of criteria, such as physical constrains and requirements (e.g., memory, processing power, passive/activeness support). In any case, the role of the TB is to complement the SD by i) retrieving from the SD information regarding user's preferences, profiles and behaviour, and consequently ii) supporting specific actions depending on the user personalization information.

A first function of the TB is to provide the user with a mean to read/write/modify the personalization information stored in the SD. From the terminal point of view, an appropriate user interface for all kinds of services has to be adapted according to the terminal's capabilities and the ambient environment (context of use). From the services and appliances point of view, a general User Interface (UI)-description language, and a common interface to all networked devices and services or methods for UI transfer to different terminals could be necessary, to simplify the development of applications.

A second function of the TB is to enable the SD to perform discovery of terminal capability and service adaptation to the ambient environment. The TB applies the user preferences to the working environment of the terminal, eventually by collecting contextual information from sensors (e.g., user location, user actions, surrounding devices and services, applications on focus…). The TB can configure applications and network settings and can dynamically download plug-ins and applications. All this information will be matched to the specific terminal capabilities in order to instantiate the proper service components on the terminal (as well as in the network, when functions such as remote trans-coding of downloaded information are necessary). Last, the TB is the entity that allows user preferences and policies, stored in the SD, to drive service adaptation to networking technologies and capabilities. Through its interface with the network, the Terminal Broker selects the most suited access alternative, and dynamically estimates current network capabilities (eventually including congestion levels, etc).

To conclude, we may say that the key feature of the TB is that all the aforementioned functions are specialized to the user actually using the involved interfaces. An important requirement of the TB is to be able to sit on top of existing and emerging terminal and network technologies. The components making up the Terminal Broker have the responsibility of decoupling the Simplicity System functions from the specific networking technologies employed and from the specific terminal characteristics. Such components might also be dynamically downloaded.

*C. Network Broker functionality*

Goal of the project is to unify and personalize the user view of available services. This implies that there must be a way to describe and advertise such services, to allow the user to browse and select them. Subsequently, there is the need to coordinate services and share/allocate available resources. A so-called Network Broker (NB) is responsible to perform the aforementioned tasks, by providing: i) a platform for service deployment, advertisement, personalization, ii) service adaptation capabilities to the considered context (location, time, etc.); iii) an orchestration of events and the handling of simultaneous access of several users to the same resources, services, and locations. the NB acts as an intelligent edge to the IP networking domain. The NB provides a network based platform. The platform takes into account user profiles, terminal capabilities as well as the network side and handles the user request in an optimized and personalized way. In order to achieve the above objectives, the NB needs to communicate with the TB in order to retrieve information about the user profile, his/her context and the device capabilities This task is responsibility of the Profile Transfer Techniques and of the Ambient Awareness Modules.

As far as the network side is concerned, the NB uses policy-based technologies (e.g., policies for mobility support, QoS, security, software downloads), exploiting a policy repository (Policies / Profiles Management Module) and a number of modules, including: AAA services module; Security module; Quality of Service / Mobility Management module; Access Technologies Management. These modules perform the orchestration of the available resources, the adaptation of network capabilities and the management of the different access technologies and networking alternatives. The NB also interacts with NBs of adjacent networks (Inter-Broker Communication module), in order to provide an optimized end-to-end service across network domains.

Another component of the NB is a platform for the deployment/advertisement of application services, which will be provided to the end user. The NB needs specifically to address the on-demand delivery of client code to user terminals. In addition, the following features are needed:

- application services providers need the possibility of deploying client components on the provisioning servers, and of associating to such components descriptive information that can be used by a client in the discovery phase;
- users need to perform discovery of software components, taking into account their preferences, contract, terminal, position, and download/install capabilities. Different methods to provide this component (e.g., MIDP OTA, JNLP) should be supported through adapters.

An attractive possible solution to complement the realization of the NB could be based on an adaptive agent-based service platform. The goal of such platform being to provide services specifically targeted to individual needs in specific environments. The platform could select these services by using information that describes the user's current environment, his or her past behaviour and the behaviour of other users in similar situations.

Another important issue is the design of a Profile Repository. This repository should provide a generic, flexible and easily updatable description of the compatible and available devices, services, technologies and users profiles of the Simplicity System. Such an overall description can be referred to as *Generic Profile or Template Profile*. *Generic Profiles* can be divided into User Generic Profiles, Devices Generic Profiles, Service Generic Profiles etc. Potential user profiles, devices, services can be categorized in groups with common features. For each one of these categories a standardized *Generic Profile* can be created and stored in the repository. For example, Business, Entertainment, Emergency Generic Profiles etc. could be possible User Generic Profiles, while PDA, Laptop, Mobile Phone Generic Profiles etc. can be the corresponding ones for the devices. From the services perspective, we can categorize services based on the different service domains they might belong to and then create the appropriate *Generic Profiles.* Such domains could be for example an enterprise network, a home network, an service provider network etc.

The transition from a Generic Profile to a specific User Profile will take place on the terminal. Each user will be able to download any of these *Generic Profiles* and modify/configure them according to his preferences. The configuration of the profile related to the devices might be done automatically with the assistance of the Terminal Capabilities Discovery module that resides within the Terminal Broker. The output of the combination of all the modified/configured generic profiles comprises the completed User Profile. Services selected by the user will be adapted according to the completed profile.

The User Profile could be stored within the SD or uploaded to the Profile Repository. In the former case, the User Profile shall be downloaded and processed by the SD on demand. In the latter case, security issues including authentication/authorization must be taken under consideration since personal data is involved. The Network Broker provides a secure, location –transparent storage service which can be used for this purpose as well as for storing session and other, application-related data on behalf of the user. Furthermore, the Profiles could be more than a mere static listing of features and capabilities. They could also embody a dynamic representation, containing both features and logic. To this end, a Mobile Agent approach could be envisaged, which will comprise the required adaptation knowledge. Mobile Agents could migrate on demand to the SD and operate in a co-operative way to formulate reconfiguration decisions that are optimal according to the user's environment. Finally, the NB architecture should be flexible enough to be used over several types of networks.

## V. CONCLUSIONS

The Simplicity project addresses a crucial issue for future systems beyond 3G and proposes a solution to handle the increasing complexity of systems, services and technologies. The concept developed in the project can directly impact the way citizens live and work. We intend to prove this concept by designing our proposed architecture and implementing its main concepts, to show its feasibility. Two companion papers provide, respectively, i) a survey of state-of-the-art candidate technologies for implementing the Simplicity architecture [3], ii) use case scenarios [4].

## VI. ACKNOWLEDGEMENTS

This work has been partially funded by the European Union in the framework of the project IST-2004-507558 Simplicity, which started on 1/1/2004. The contribution of all Simplicity partners to this paper is gratefully acknowledged. However, that support should not be construed as an endorsement of the views, results and conclusions contained in this paper, for which the Authors are wholly responsible.

## REFERENCES

[1] Website of the Simplicity project: http://www.ist-simplicity.org

[2] C. Noda et al. : "Distributed Middleware for User Centric System", WWRF#9 conference, Zurich, July 2003.

[3] N. Blefari-Melazzi et al.: "The Simplicity Project: easing the burden of using complex and heterogeneous ICT devices and services. Part II: State of the Art of Related Technologies", IST Mobile&Wireless Communications Summit 2004, June 27-30 2004, Lyon, France.

[4] R. Seidl et al.: "The Simplicity Project: Personalized and simplified communication spaces for mobile users", IST Mobile&Wireless Communications Summit 2004, June 27-30 2004, Lyon, France.