Specification of context-aware mobile services over a distributed brokerage framework: The ubiquitous Media Streaming example

E. Koutsoloukas[∇], S. Kapellaki, J. Papanis, N. Dellas, N. Tselikas, G. N. Prezerakos[+], D. I. Kaklamani, I. S. Venieris

National Technical University of Athens, School of Electrical & Computer Engineering,
9 Iroon Polytechniou Str., 157 73 Athens, Greece
E-mail: {lefterisk, sofiak, jopapan, ndellas, ntsel, prezerak}@telecom.ntua.gr, dkaklam@mail.ntua.gr, venieris@cs.ntua.gr

Abstract: **The paper targets the specification process of context-aware end-user services targeted at mobile systems beyond 3G based on two key concepts. A personalization device that enables services to adapt to a variety of terminals, networks and user preferences and a distributed brokerage framework that orchestrates service related resources by implementing adaptive policies for inter-broker message exchange. Both concepts are discussed in detail via a real-life paradigm, the high level specification of a personalized context-aware media streaming service.**

1. INTRODUCTION

Systems beyond 3G are characterized by a multitude of services offered over a variety of network and terminal technologies. In many cases, the resources required for service provision are distributed among various network nodes and corresponding physical locations. This situation results in increased complexity in the specification, implementation as well as the operation of such services [1].

On the other hand, the resulting complexity must remain transparent to the end-user for end-user services to become commercially attractive. Moreover, services offered should be secure, Internet-able, mobile, personalized and context-aware according to the global Internet vision that calls for an invisible infrastructure that will allow anyone to plug in from any location with any device at any time [2].

The paper begins by outlining the approach towards ubiquitous service provision that is followed within the SIMPLICITY IST project [3]. Obviously, specification of services and the corresponding network architectures is the starting point of any design regardless of the preferred development methodology. Section 3 explains in detail the specification of a communication mechanism between the broker entities of a distributed brokerage framework that enables the coordination of the various service related components. Section 4 discusses the specification of media streaming service and shows how the service specification process takes into account the personalization and context-awareness facilities offered by the framework in question. Finally the paper delves into the relation between the specification process and subsequent steps towards service implementation.

2. THE SIMPLICITY APPROACH

The IST SIMPLICITY project (Secure, Internet-able, Mobile Platforms LeadIng CItizens Towards simplicity) aims at users of existing as well as emerging services who employ different terminals (including PCs, PDAs and mobile phones), over wired and wireless networks. The user should enjoy personalised context aware services with minimal configuration overhead. With this in mind, the high-level description of the project's architecture [4] includes two key concepts – the Simplicity Device (SD) and the Brokerage Framework.

The SD represents the access key to every feature and service of the Simplicity system. Each user will be equipped with a personal SD, which will customize the available terminal according to the user's profile and preferences and will allow uniform and secure access to heterogeneous networks and services. Users will be able to edit their preferences at will. In addition, preferences will be dynamically auto-updated by taking into account the user's behaviour. Another interesting SD feature is session saving/transferring, allowing the user to continue his/her work with a different terminal device without losing time for reinstalling the needed applications.

The Brokerage Framework provides the means to use network capabilities such as mobility control, QoS, security and adapt them to user preferences and terminal characteristics in a manner transparent to the user. Policy-based mechanisms will control the flow of messages between the different system components.

The approach followed in SIMPLICITY is expected to facilitate Value Added Services (VAS) provision since

[∇] *Corresponding author.*

[+] *Dr. Prezerakos is also with the Technological Education Institute (TEI) of Piraeus, Dpt. of Electronic Computer Systems, 250 Thivon Av. & Petrou Ralli, 122 44 Athens, Greece, e-mail: prezerak@teipir.gr*

SIMPLICITY-aware networks will automatically inform the users of available services, adapt them to the underlying network infrastructure and terminal device and ease the access process (e.g. by performing automatic user authentication).

3. AN INTERBROKER COMMUNICATION MECHANISM

The use of brokers as orchestrators of network resources [5] as well as for supporting multimedia and context-aware systems [6], [7] is a widely used concept. The core architecture of SIMPLICITY system is based on an evolution of a brokerage framework [7] consisting of a set of brokers that interact with each other and exchange information. Each SIMPLICITY broker consists of a set of subsystems that implement the service logic that the broker hosts. The subsystems are connected through a mediator entity that implements an event-based intra-broker communication mechanism. The mediator collects the events that subsystems generate and redistributes them to other subsystems based on a predefined set of policy rules. The distribution of events is local, in the sense that the mediator only distributes events to the subsystems that are attached directly to it.

For the inter-broker communication mechanism, a special subsystem called SIMPLICITY Broker Communication Subsystem (SBC) is introduced. The SBC is responsible to orchestrate every aspect of the interaction between subsystems that are attached to different mediators. The SIMPLICITY brokerage architecture is illustrated in the figure below:

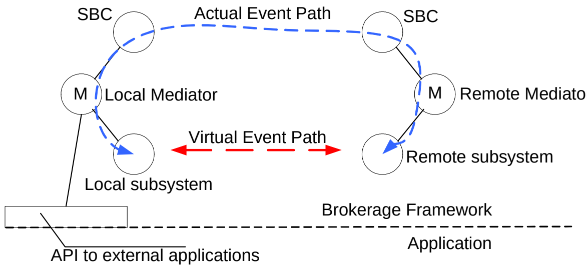

Figure 1 - Broker architecture and the SBC subsystem

The SBC subsystem's role in this architecture is to extend the event-based intra-broker communication to include remote subsystems as well, in a way transparent to the local and remote mediator entities. When a subsystem expresses the interest to receive events generated by a remote subsystem, attached to a remote mediator, a negotiation is triggered between the local and remote SBC subsystems. Whenever the event is generated remotely the remote SBC delivers the event to the local SBC, using a custom or standardized XML based serialization protocol [8] and the event is regenerated locally and delivered to the local subsystem by the local mediator.

Prior to this operation, a discovery phase takes place, during which each SBC subsystem discovers other brokers, the subsystems they host and the events they generate, which are described in a formal XML based description language, such as WSDL [9]. The discovery phase may be assisted by peer-to-peer technology, so that the SIMPLICITY brokerage framework can operate as a decentralized system, with significant gain in flexibility and scalability.

Therefore, the overall SIMPLICITY brokerage framework can be seen as a very flexible publish/subscribe system [10], which features a registry of publishers and subscribers distributed throughout the system and whose asynchronous and scalable nature allows a quick adaptation to changing networking environments, such as the ones that include unreliable wireless links [11], [12]. The graph of interconnections between subsystems is adjustable and flexible and the result is a loosely coupled set of subsystems, each one implementing a specific part of service logic and cooperating to deliver a higher-level service to applications.

The Simplicity Device (SD) is incorporated into this framework as yet another subsystem. The SD may come in different flavours, ranging from a processing and storage equipped device to a simple memory card to a virtual network entity. In all cases it interfaces the SIMPLICITY architecture via the predefined communication mechanisms and provides authorization and personalization facilities to end-user services.

The interaction between applications and the SIMPLICITY brokerage framework is done through an API that hides the brokerage and event-based nature of SIMPLICITY and focuses on a set of high-level services that SIMPLICITY offers. Such services would be downloading and uploading files, saving and restoring application sessions, requesting specific Quality of Service, requesting AAA services, managing complex user contexts etc. The SIMPLICITY API to applications takes the form of method calls to a generalized "SIMPLICITY" entity, which takes the responsibility to orchestrate subsystems and events in order to deliver the requested service to the application. Therefore, application developers do not need to be introduced to the complex and dynamic nature of the SIMPLICITY system and they can focus on enriching their applications with features that ease the user's interaction with the complexity of the modern ICT environment.

4. SPECIFICATION OF A UBIQUITOUS MEDIA STREAMING SERVICE

The component communication framework is specified in parallel to the actual end-user service specification. SIMPLICITY plans to implement and demo several end-user services which are specified via UML constructs.

The next section presents a functional view of the way by which the SIMPLICITY concepts (SD and communication framework) are employed in the specification process of a SIMPLICITY-aware “Buy, Store & Stream Media” service.

4.1 Narrative Scenario Description

A scenario involving a user subscribed to an online service in order to “Buy Music” is examined. The user inserts his SD in his PC at home and after the initial procedure of login takes place, a dialog appears informing the user about the services that he is able to access. He decides to use the on line “Buy Music” service. He selects a few songs from his favourite band's latest album. The service retrieves from the SD his credit card number and the information related to the network data storage where he usually saves the songs he buys. The user confirms the payment and the storage of the songs. Consequently, the audio files are copied to his SIMPLICITY personal space on a network disk and his play-list is updated. The songs are available to him anytime anywhere from any SIMPLICITY-enabled device that supports streaming audio, RealAudio for instance. The user logs off the service.

Later that day, while being at a coffee shop that provides WLAN Internet access, the user decides to enjoy his coffee while listening to his favourite band. So he connects his SD to his PDA and the log in procedure takes place. From the available services presented to him, he launches the Simplicity Media Streaming Service. The service retrieves the play list from the network data storage and presents it to the user. He selects the recently bought songs and they are then streamed to his PDA using the appropriate streaming technology according to the PDA's capabilities and the available bandwidth. In the process, a streaming audio plug-in may be required to be downloaded and installed.

4.2 Analysis - UML Diagrams

The analysis of the narrative Media Streaming scenario involves the separation of the scenario in Use Cases (UC), the identification of the different actors that participate and the interactions between them, as well as the required SIMPLICITY functionalities. Three different use cases were identified, UC1: SD Plug in Procedure, UC2: Buy media on-line and UC3: Enjoy Streamed Media. The directly related to the Media Streaming Service use cases are UC2 and UC3, while UC1 is a basic use case that takes place every time a user wants to access the SIMPLICITY environment.

Also the following actors were identified:

- Simplicity Device (SD)
- Terminal/Simplicity Terminal Broker (STB)
- Simplicity Network Broker (SNB)
- Simplicity Service Broker (SSB)
- Network Provider
- Value-added Service Provider (VASP)

The Simplicity Terminal Broker (STB), the Simplicity Network Broker (SNB) and Simplicity Service Broker (SSB) comprise entities of the SIMPLICITY Brokerage Framework, each one offering specific functionalities. The Brokers interact with each other, as well as with the SD and with external actors such as the Network and the Service Providers.

The high-level UML Sequence diagrams that follow provide in a graphical way the results of our analysis.

4.2.1. UC1: SD Plug in Procedure

The user inserts the SD to his terminal. The Simplicity Terminal Broker (STB) residing in it takes action in order to authenticate the user to the SIMPLICITY environment. An appropriate dialog prompts the user to insert his username and password and the STB interacts with the SD in order to crosscheck them. After the authentication procedure, the STB gathers user’s preferences, profiles and general personal data from the SD and also discovers the terminal’s capabilities. Based on the gathered information the TB performs personalization actions related to the terminal and the network connection, e.g. customization of the working environment, selection of the best available network according to user’s preferences.

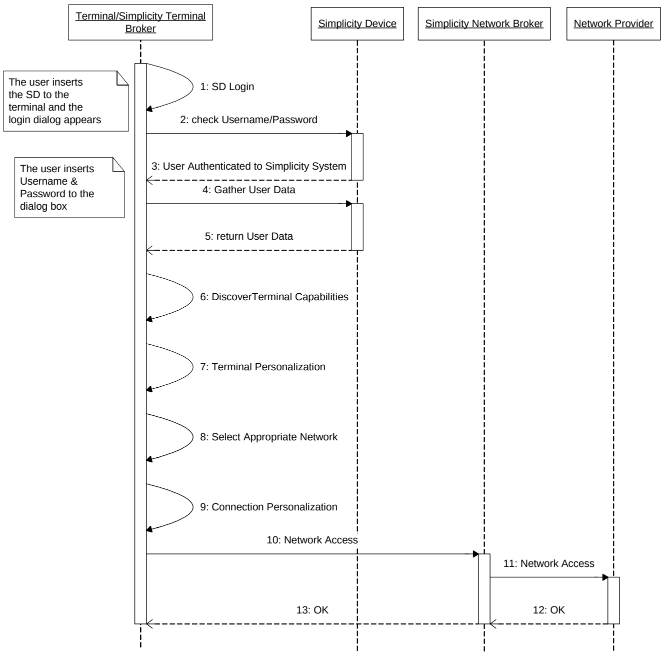
Terminal/Simplicity Terminal Broker
Simplicity Device
Simplicity Network Broker
Network Provider
The user inserts the SD to the terminal and the login dialog appears
1: SD Login
2: check Username/Password
The user inserts Username & Password to the dialog box
3: User Authenticated to Simplicity System
4: Gather User Data
5: return User Data
6: DiscoverTerminal Capabilities
7: Terminal Personalization
8: Select Appropriate Network
9: Connection Personalization
10: Network Access
11: Network Access
13: OK
12: OK

Figure 2 – UC1: SD Plug in Procedure

Consequently, the STB communicates with the Simplicity Network Broker (SNB) requesting access to the selected network. Network Authentication is performed with the collaboration of the different Simplicity entities (STB, SNB) and the Network Provider transparently to the user.

4.2.2. UC2: Buy media on-line

Now that the user has obtained network connectivity, the STB interacts with the Simplicity Service Broker (SSB) in order a list with the available services to be returned to the user. The SSB acts as an intermediate between the user and the Value-added Service Providers (VASP). The user selects the service he wants to access. The STB retrieves from the SD authentication data, passes it to the SSB, which authenticates the user to the service by interacting with the VASP domain and the service is available to the user via a personalized User Interface. All he has to do now is select the media he wants to buy. Afterwards, with the collaboration of the different Simplicity Brokers and the SD user's credit card number and his network storage location are provided to the service. The songs are purchased and copied to the user's storage in the network. All these actions are performed automatically. The user just confirms them.

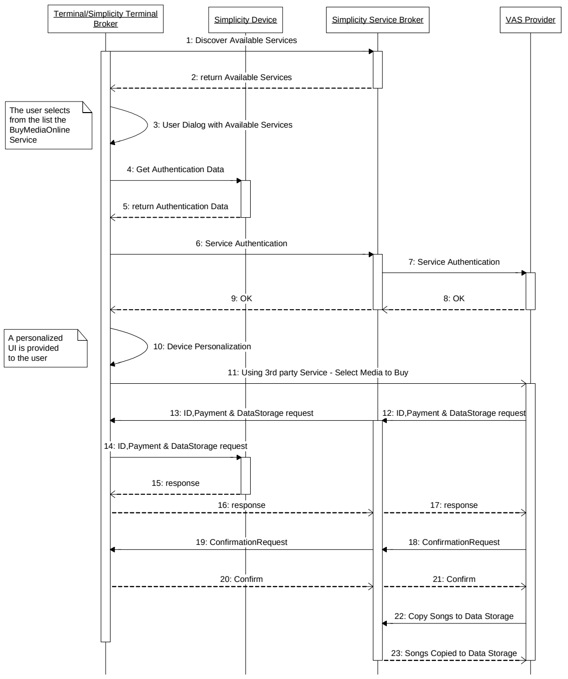
Terminal/Simplicity Terminal Broker
Simplicity Device
Simplicity Service Broker
VAS Provider
1: Discover Available Services
2: return Available Services
The user selects from the list the BuyMediaOnline Service
3: User Dialog with Available Services
4: Get Authentication Data
5: return Authentication Data
6: Service Authentication
7: Service Authentication
9: OK
8: OK
A personalized UI is provided to the user
10: Device Personalization
11: Using 3rd party Service - Select Media to Buy
13: ID,Payment & DataStorage request
12: ID,Payment & DataStorage request
14: ID,Payment & DataStorage request
15: response
16: response
17: response
19: ConfirmationRequest
18: ConfirmationRequest
20: Confirm
21: Confirm
22: Copy Songs to Data Storage
23: Songs Copied to Data Storage

Figure 3 – UC2: Buy media on-line

4.2.3. UC3: Enjoy Streamed Media

At this point, the user can access his songs anytime anywhere from any SIMPLICITY enabled terminal provided that the SD login procedure is completed. He just has to select the Media Streaming Simplicity Service from the list of the available SIMPLICITY services. In this case there is no need for service authentication through the SSB like in UC2. The user can exploit all SIMPLICITY services from the moment he is authenticated to the SIMPLICITY environment during the initial SD login procedure. The STB accesses through the SNB the user's personal data storage and presents to the user through a personalized User Interface the available songs. The user performs the selection of the songs he wishes to listen. The next step is the interaction of the Simplicity Brokers with the Network Provider domain for the retrieval of information regarding the available bandwidth. If the offered bandwidth meets user's requirements, the content is properly adapted and streamed to the user's terminal.

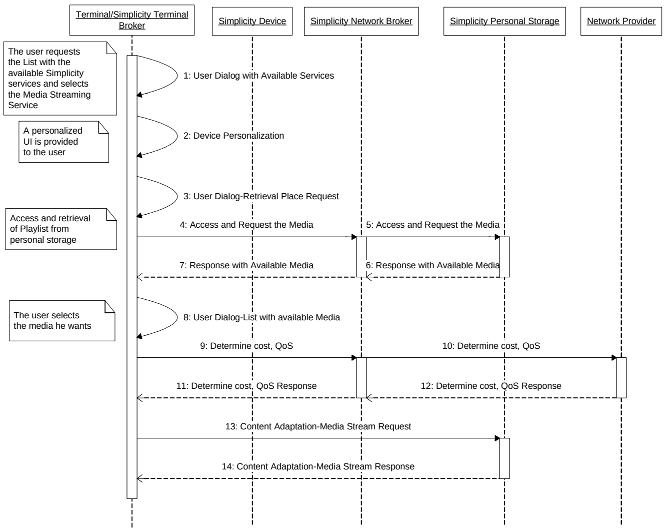
Terminal/Simplicity Terminal Broker
Simplicity Device
Simplicity Network Broker
Simplicity Personal Storage
Network Provider
The user requests the List with the available Simplicity services and selects the Media Streaming Service
1: User Dialog with Available Services
A personalized UI is provided to the user
2: Device Personalization
3: User Dialog-Retrieval Place Request
Access and retrieval of Playlist from personal storage
4: Access and Request the Media
5: Access and Request the Media
7: Response with Available Media
6: Response with Available Media
The user selects the media he wants
8: User Dialog-List with available Media
9: Determine cost, QoS
10: Determine cost, QoS
11: Determine cost, QoS Response
12: Determine cost, QoS Response
13: Content Adaptation-Media Stream Request
14: Content Adaptation-Media Stream Response

Figure 4 – UC3: Enjoy Streamed Media

5. CONCLUSION AND FUTURE WORK

SIMPLICITY builds on the concepts of personalization devices in conjunction to a distributed brokerage framework for enhancing user experience for many context-aware applications, in different network and terminal environments. Based on the example of a ubiquitous media streaming service, we have depicted how a thorough analysis that identifies actors and their interactions followed by the identification of specific use cases can exploit these concepts to their full extent. Our final goal is to provide a secure, Internet-able, mobile and personalized framework for context-aware applications and especially for media streaming. With this in mind, the project plans to continue to work in a number of different directions.

With respect to implementation of the personalization device, we are experimenting with two categories of Simplicity Devices, JavaCards and Java-enabled SIM cards [13]. For the access part we also plan to employ a number of wireless access technologies such as Bluetooth and wireless LANs. Also, we are currently evaluating various peer-to-peer frameworks for possible use in the broker-to-broker communication mechanism.

The QoS parameters that should be provided by the SIMPLICITY components involved in creation and delivery of applications are also under investigation. Moreover, policy-based decisions [7], [14], [15] and content adaptation are considered as additional factors of the project's success. In the future, more context-aware mobile applications operating under different types of context information will be examined.

Another important aspect of the project is the area of performance evaluation. Performance will be evaluated with respect to different use cases, alternative distributions of components and use of different inter-broker communication protocols.

6. ACKNOWLEDGMENTS

This work has been partially funded by the European Union in the framework of the project IST-2004-507558 Simplicity (http://www.ist-simplicity.org), 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] S.Y. Hui, K.H. Yeung, "Challenges in the Migration to 4G Mobile Systems", IEEE Communications, Vol.41, No. 12, Dec. 2003, pp. 54-59.

[2] L. Kleinrock: "An Internet vision: the invisible global infrastructure", Ad Hoc Networks, Volume 1, Issue 1, July 2003, pp. 3-11.

[3] 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", IST Mobile & Wireless Communications Summit 2004, Lyon, France, June 2004.

[4] 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, Lyon, France, June 2004.

[5] Klara Nahrstedt, Jonathan M. Smith, "The QoS Broker", IEEE Multimedia, Vol. 2, Issue 1, March 1995.

[6] H. Chen, T. Finin, A. Joshi: "An Intelligent Broker for Context Aware Systems", Adjunct Proceedings of Ubicomp 2003, Seattle, Washington, USA, October 2003.

[7] Kassler, A. Schorr, L. Chen, C. Niedermeier, C. Meyer, M. Helbing, M. Talanda: "Multimedia Communication in Policy based Heterogeneous Wireless Networks", IEEE Vehicular Technology Conference VTC2004-Spring, Milan, Italy, May 2004.

[8] Simple Object Access Protocol Specifications: http://www.w3.org/TR/soap/

[9] Web Service Description Language Specifications: http://www.w3.org/TR/wsdl

[10] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraui, Anne-Marie Kermarrec, "The many faces of publish/subscribe", ACM Computing Surveys, Vol. 35, Issue 2, June 2003.

[11] Gianpaolo Cugola, H.-Arno Jacobsen, "Using Publish/Subscribe middleware for mobile systems", ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 6, Issue 4, October 2002.

[12] Yongqiang Huang, Hector Garcia-Molina, "Publish/Subscribe in a mobile environment", Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access, Santa Barbara, California, USA, May 2001.

[13] JavaCard Technology website:
http://java.sun.com/products/javacard/index.jsp

[14] J. Keeney, V. Cahill: "Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework", Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, June 2003.

[15] P. Lago: "A Policy-based Approach to Personalization of Communication over Converged Networks", 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02), Monterey, California, 2002.