RECONFIGURATION MANAGEMENT IN 4G MOBILE NETWORKS: REQUIREMENTS, PROCESS AND ARCHITECTURE

Vangelis Gazis, Nikos Houssos, Nancy Alonistioti, Lazaros Merakos
Communication Networks Laboratory, University of Athens, Athens, Greece
e-mail: {gazis@di.uoa.gr, nhoussos@di.uoa.gr, nancy@di.uoa.gr, merakos@di.uoa.gr}

ABSTRACT

Global vision consensus on the next generation of wireless mobile communications, broadly termed 4G, sketches a heterogeneous infrastructure, comprising different wireless access systems in a complementary manner and vested with reconfiguration capabilities that will facilitate a more flexible and dynamic adaptation of the wireless network infrastructure to better meet the ever-changing service requirements. In the present paper, we analyze and identify the functional requirements of reconfiguration, presenting a generic staged reconfiguration procedure and identifying the management requirements in the context of these stages. Next, we introduce a generic, reconfiguration management framework to be hosted on 4G mobile systems and elaborate on its main components.

1. INTRODUCTION

Fourth generation wireless (4G) will be mainly characterized by a horizontal communication model, where multiple different access technologies such as cellular, cordless, wireless LAN, short range connectivity and even wired systems will interface to a common platform over the IP protocol, complementing each other in an optimal way for different service requirements and radio environments. Supported by their personal intelligent agent(s), users who roam in the 4G 'technological mosaic' will enjoy 'always best' connectivity and seamless access to value-added services, third-party applications and content over the most efficient combination of systems available [1]. Considering that the observed proliferation of wireless technologies is likely to persist and that future mobile equipment will have to support multiple dissimilar wireless standards, the mobile industry has been focusing on reconfigurability as a technological enabler of future multi-standard mobile systems and flexible joint radio resource management across different wireless standards. The rest of the paper is structured as follows: Section 2 highlights the fundamental concepts of reconfigurability, providing key definitions for the next sections. Section 3 discusses reconfigurability and service provision issues in the 4G era and presents a service provision platform. Section 4 identifies the main functional requirements of reconfiguration and introduces a generic staged reconfiguration procedure Section 5 introduces the proposed reconfiguration management framework and its main functional components. Finally, Section 6 concludes the paper and provides directions for future work.

2. RECONFIGURATION: CONCEPTS & ISSUES

Over the last few years, a number of EU research projects in mobile communications (FIRST, SORT, TRUST, CAST, MOBIVAS, SCOUT) have addressed the thematic area of reconfigurability, identifying main functional requirements and proposing a number of reconfiguration-supporting architectures, each tailored to fit the problem domain under study by each project. However, although the volume of reconfigurability-related research contributions is constantly increasing, the paramount issue of reconfiguration control and management has remained in twilight. Naturally, the broader issue of a generic, all-encompassing reconfiguration management architecture remains an open one.

Scope and definition

Reconfigurability deals with issues such as the dynamic instantiation, parameterization and inter-connection of functional entities (e.g., protocols) within the user, control and management planes of a collection of operating communication systems in a manageable, consistency-preserving and – preferably – transparent fashion. Supported by over-the-air download of required system software an/or firmware, reconfigurability possesses a huge potential for the design of ubiquitously adaptable and, more importantly, evolvable mobile systems. By exploiting over-the-air download and in-situ installation of appropriate communication protocols and/or middleware components and instantiating appropriate communication personalities (i.e., user, control, or even management planes) within its target devices, reconfigurability enables a very dynamic and flexible adaptation of the communication equipment's behavior to optimally meet the instant service requirements. For the rest of the present paper, the term reconfiguration

This work has been partially performed in the framework of the project "ANWIRE" (www.anwire.org) funded by the European Community under contract IST-2001-38835

will refer to the dynamic adaptations of implementation mappings of communication components [2] that do not compromise their consistency or their ability to continue to provide services.

Related work

SDR Forum pioneered in producing a Software Radio Architecture (SRA) specification [3] providing a framework for building, configuring, connecting and tearing down distributed, embedded (radio) applications within a radio device. SRA specifies software (OMG IDL) interfaces for the installation and use of distributed (radio) applications within a single device so as to support reprogrammable communication capabilities. It uses a variant of the CORBA Components specification to describe the hardware and software components of an SDR system, their properties and their interconnections. Building on and expanding the work of the SDR Forum, the OMG Software Radio (SWRADIO) Special Interest Group (SIG) is primarily focusing on the development of an architecture that will allow applications to be independent from the operating environment within a radio, thereby being portable across a variety of hardware configurations. Notably, it recognizes that such portability requires hardware interfaces to be abstracted and application management interfaces to be standardized along with common metadata and loading processes [4]. In brief, the majority of existing work on architectures for reconfigurable communication systems focuses mostly on the architectural (i.e., structural) view and its constituent functional interfaces and does not address other, equally important concerns, such as generic decision support for optimal reconfiguration sequences, or what the generic phases of reconfiguration procedures are.

3. SERVICE PROVISION AND RECONFIGURATION

Service provision issues

The ubiquitous provision of (mobile) services, over the dynamically reconfigurable networking and computing infrastructures that will be commonplace in the beyond 3G mobile era, requires a common set of support mechanisms that, based on each application's requirements, identify and trigger the appropriate reconfiguration actions on the underlying network infrastructure, including the mobile devices themselves, in a manner that is transparent to the user. Besides structural concerns dealing with the internal re-organization of affected communication systems, these mechanisms must also address issues related to context capture and discovery, capability negotiation procedures and representation formats, decision support regarding the optimal communication mode, over-the-air deployment of software implementations (e.g., MAC protocols, codecs, modulation components, etc), in-situ installation and validation checks, as well as remote service management (e.g., activation/deactivation of protocol instances within a mobile device). Obviously, the ultimate distribution of the functionality in support of these mechanisms will depend on the overall service provision architecture.

Service provision platforms that mediate between network equipment, including end-user devices, and applications have been at the focus of the EU IST project MOBIVAS, which validated the case of a service platform for over-the-air provision of applications in 3G and beyond 3G mobile environments. The next section introduces the MOBIVAS architecture and presents its main components.

The MOBIVAS platform

The MOBIVAS platform addresses major issues regarding deployment and management of independently developed applications targeting the users of 3G and beyond 3G mobile networks. Applications are typically contributed by third-party software vendors, termed Value-Added Service Providers (VASPs) and must be delivered over different types of network infrastructures that may be operated by different administrations. To support deployment over a wide range of possible end user devices, the value-added service (VAS) architecture consists of two components: A "server" component hosted on the VASP's application servers and one or more "proxy" components, each tailored to a particular execution environment (e.g., J2ME, Personal Java, etc) and/or mobile device capabilities (e.g., cellular phone, PDA, etc). While the server component is regarded as stationary, the proxy component is downloaded and executed at the mobile device upon the (user's) request. The logical architecture of the platform is depicted in the center of Figure 1, which also provides a high-level visual of the platform's deployment in a 3GPP UMTS network. The main logical components of the platform are:

The *Value-Added Service Manager* (VASM) is the central platform component in that it co-ordinates the entire service provision and management process. It includes modules that undertake on-line VASP-initiated service registration and deployment, including any network reconfiguration that may be necessary, maintenance of service and user metadata in suitable databases and repositories, as well as facilities for customized online service discovery, downloading and adaptation.

The *Charging, Accounting and Billing* (CAB) system is responsible for producing a single user bill for service access and apportioning the resulting revenue between the involved business players.

The *End User Terminal Platform* (EUT) includes functionality such as service downloading management, GUI clients for service discovery and selection, capturing of event notifications and service execution management.

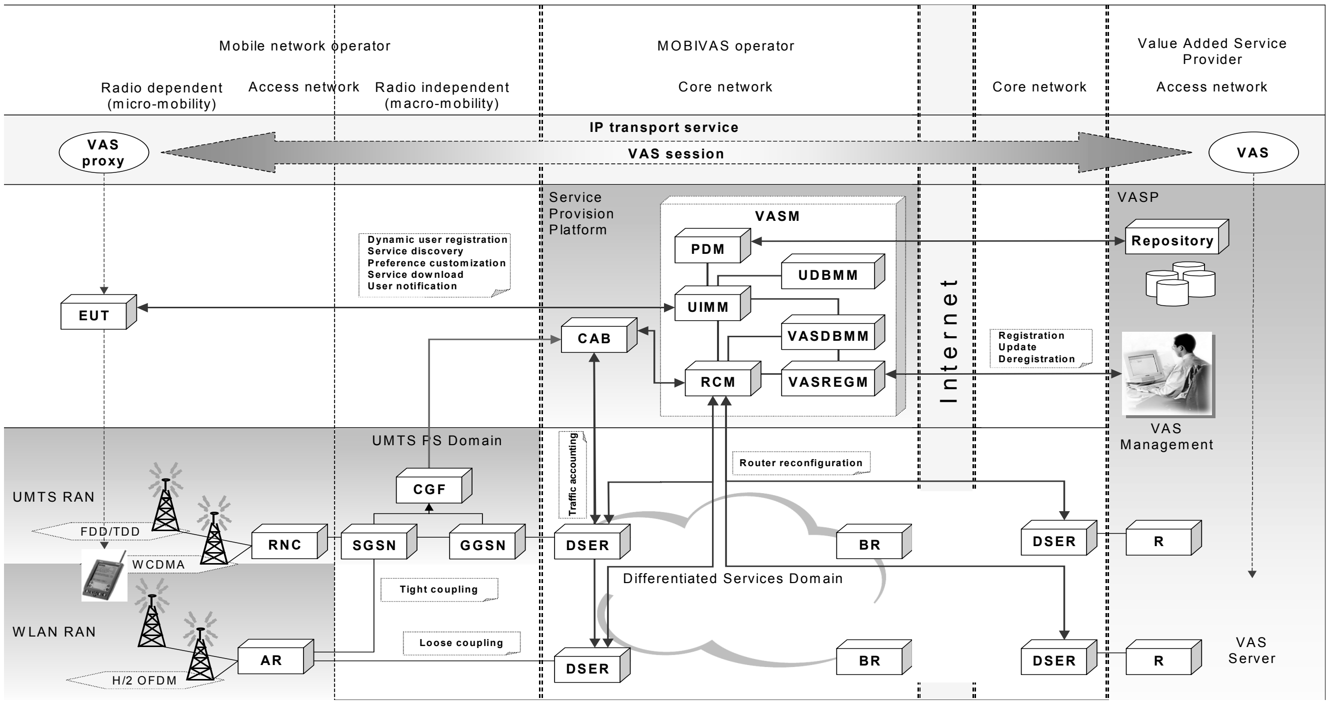

Figure 1. Deployment of the MOBIVAS service provision platform in a 3GPP UMTS R5/R6 environment.

The *DiffServ Edge Router* (DSER) provides for QoS-aware traffic classification by hosting DiffServ edge functionality, traffic monitoring (i.e., accounting in IETF parlance). It provides detail records to the CAB, regarding the traffic patterns of deployed value-added services and can be reconfigured dynamically with regard to the employed QoS classification rules and DiffServ classes as well as the traffic monitoring and reporting functionality.

Focusing on the internal VASM architecture, we present its main functional components:

The *VAS Registrar Module* (VASREGM) dealing with VAS registration and deployment. Its functionality includes storing the VAS profile that records VAS metadata for QoS, charging and location aspects in an XML format in the VAS database and also performing any necessary reconfiguration actions on network elements (e.g., DSER).

The *User Interaction Management Module* (UIMM) that co-ordinates all user-centric operations of the platform (e.g., authentication procedures, service discovery and selection and secure downloading) while enhancing them with flexible adaptation and context-aware features.

The *Packaging and Downloading Module* (PDM), which is responsible for packaging all the software components and other supporting resources (e.g., images, etc.) required for executing a VAS in a single archive. The single archive produced is dynamically tailored to the context of the particular VAS execution request (e.g., terminal and network characteristics, user preferences).

The *User Database Management Module* (UDBMM) that handles all user-related metadata profiles to enable service discovery, adaptation and provision according to user preferences. The user profile contains information such as user identification data (e.g., name, IMSI, security keys), generic, service-independent preferences (e.g., language, default tariff class), user interface preferences (e.g., font size), as well as references to user-specific favorite services.

The *VAS Database Management Module* (VASDBMM) that provides for back-end persistence of user, terminal, service and network metadata.

The *Reconfiguration Manager* (RCM), an extension of the original MOBIVAS design, undertakes network, and service reconfiguration, applying proper reconfiguration procedures upon network elements and/or activating a generic service adaptation mechanism with intelligent profile matching. In the next section, we focus on the design aspects of the RCM.

4. THE RECONFIGURATION MANAGER (RCM)

Basic reconfiguration concepts

Treating reconfiguration as the selective adaptation of implementation mappings for modular (i.e., component-based) subsystems, we consider the following classification for reconfiguration-related mechanisms:

a) Mechanisms that realize functionality prerequisite to the reconfiguration action per se and are orthogonal to it, e.g., context-capture functionality, device capability negotiation, intelligent decision making, over-the-air download, etc.

b) Mechanism that support the reconfiguration action per se, i.e., functionality related to the instantiation, inter-connection, activation and/or deactivation of communication components within their execution environment (e.g., in the case of protocols, the device's communication subsystem).

Typically, the latter category of reconfiguration-related functionality will be available on each communication system that may be subjected to reconfiguration, while the former can be factored out to the service provision platform that undertakes the bulk complexity of pre-reconfiguration actions, including necessary interactions with any affected (proximal) network elements. This approach, simplifies the design of reconfigurable equipment by outsourcing certain reconfiguration issues to the service provision platform, where, by taking advantage of the abundant resources in the wireline infrastructure, and the much broader contextual information available (e.g., network capabilities, traffic load of base stations proximal to the mobile device, etc), more accurate decision-making about the optimal communication modes and/or the best reconfiguration strategy for a given mobile device can be made. Naturally, security issues are paramount and, in such a case, would require that the platform is administered by a party trusted by the owners of reconfigurable equipment (e.g., a regulatory body).

Regarding the reconfigurable communication system realm, we advocate a component-based (and thereby modular) architecture and describe a generic reconfiguration management framework as a natural, complementary extension of the service provision platform to realize a comprehensive, unified reconfiguration management framework. In addition, we propose that network equipment, mobile or not, be enhanced with a *generic capability for reconfiguration*, i.e., the ability to dynamically change the implementation mappings of their constituent (functional) components. Fundamentally, that requires research and standardization in two key areas:

a) Object-oriented information models [5] to capture and express the internal organization and structure of network equipment in an abstract, implementation neutral way that effectively provides the unified view necessary to start specifying the generic change capability. Through object orientation and inheritance, common structural parts can be factored out and reused as an abstract information model from which wholly different structures will inherit their common parts, allowing a fine-grain mix of standardized behavior with innovative, performance-focused, proprietary instrumentations.

b) Reconfiguration-related behavioral semantics recorded within a platform-independent reconfiguration management framework that specifies reconfiguration-enabling functional interfaces and the blueprints of their interactions (e.g., message sequence charts, state machines diagrams, etc) so as to support generic, technology-independent reconfiguration procedures.

A generic staged reconfiguration procedure

Based on the above classification of reconfiguration-related mechanisms, we postulate that reconfiguration procedures progress sequentially through the following stages:

a) *Context discovery and identification*: As reconfigurability deals with the dynamic adaptation of implementation mappings for functional communication components, it requires contextual awareness, i.e., being able to discover and exploit contextual information useful in manifesting certain behavior [6]. For example, consider the hypothetical case of a mobile device equipped with a reconfigurable communication subsystem under WLAN and GSM/UMTS radio coverage. From the prism of reconfiguration, context identification will identify which wireless network elements (e.g., Node B, RNC, BTS, BSC, WLAN access point) are capable of participating in reconfiguration-related signaling. This discovery process considers the immutable capabilities of network equipment (e.g., hardware revision levels, execution environment features, supported communication standards, etc) and identifies the particular mobile network elements that can be reconfigured in the current context. To achieve this, common information models that express the functional mechanisms of equipment components in a universal manner independent of any implementation artifacts are necessary while industry-standard flexible representation formats (e.g., XML) can be employed to encode such information models in an interoperable way.

b) *Discovery of feasible communication personalities*: The set of candidate network elements is used to identify the initial set of possible communication personalities (i.e., standards) on the basis of necessary hardware and software capabilities. This initial set is further refined by ruling out combinations of communication personalities for which equipment resources do not suffice. Thus, this phase interacts with resource management to consider dynamic equipment capabilities (e.g., buffering capabilities, traffic load, processor utilization level, etc) as filtering criteria.

c) *Decision of optimal communication personalities*: The need for personalization throughout the 4G mobile service provision process [1] requires that user preferences must be accounted for when choosing the optimal personalities for personal mobile devices. For instance, users may wish to maximize the available communication options of their mobile device (i.e., the set of communication standards it supports) or prefer a particular wireless access standard (e.g., WLAN) to others. To support arbitrary selection criteria and decision-making algorithms, reconfiguration management must furnish open interfaces to input the decision algorithm's data, decision variables and criteria values, to access its output results and to support switching between different decision algorithms during runtime [7].

d) *Deployment of selected communication personalities*: Having identified the set of optimal communication personalities, reconfiguration will proceed to download the appropriate implementations from their online software repositories. Once the download completes, implementation classes are instantiated in main memory, intra-system inter-protocol communication links (e.g., IPC, message queues, shared memory, etc) are established and the protocol graphs of the chosen communication personalities are built. Finally, protocol instances are activated (i.e., given their own thread of control) and register their Service Access Points (SAP) for use by dependent intra-system protocols.

The last (i.e., the deployment) stage identifies the generic functionality to download, install, activate – and also deactivate, uninstall and remove – functional components within communication systems; functionality which must be resident in all reconfigurable systems and ideally, constitute part of a broader management framework. The next section elaborates on the proposed such management framework.

5. A RECONFIGURATION MANAGEMENT FRAMEWORK

The reconfiguration management framework targets mainly the subject of reconfiguration procedures (e.g., the DSER and/or the EUT in Figure 1) and has been specified in UML, thereby being platform-independent. The key benefit of a platform independent model (PIM) standard is portability to different platform specific models (PSM) such as CE, .NET, CORBA, or Java using standardized mappings. Another major benefit is the ability to certify compliance of the PSM as it is mapped to the PIM using standard technologies such as UML and XML which support formal methods to prove compliance of a particular PIM-PSM pair, provided the mappings themselves are done with formal methods in mind. This modeling approach makes it possible to design an open architecture that embraces the entire infrastructure, from the end-user's mobile device through the network equipment, including proximal mobile devices in the case of ad-hoc and self-organizing networks, to value-added services and other the applications.

Framework design principles

Our proposed framework is founded upon a set of UML management interfaces that support asynchronous command dispatching to the managed objects and asynchronous notifications from those objects towards their managing entity. The reconfiguration management functionality is specified on top of this basic 'command and notification' functionality, following a 'separation of concerns' principle that discriminates between orthogonal concerns, thereby leading to modular and extensible design (Figures 2 and 3).

Reconfiguration management functionality results through the composition – rather than the inheritance – of the specialized interfaces (e.g., LifecycleManagement), thereby facilitating parallelism in the design of reconfiguration signaling. For example, deactivation of an existing protocol and download of a new implementation for that particular protocol may progress simultaneously, because the inherited management functionality provides for the asynchronous interactions needed in concurrent operations while composition (as opposed to inheritance) facilitates the use of multiple instances that each employ asynchronous interactions in its own context. Figure 4 illustrates the interfaces employed by the reconfiguration management framework classes and interfaces, which are further explained in the following section.

Reconfiguration management framework interfaces

The *System Registry* (not shown in Figure 4) is the core of the reconfiguration management framework by providing a single reference point for management and discovery of globally available reconfigurability meta-information, such as equipment hardware versions, communication subsystem capabilities, protocol implementation versions, execution environment properties, etc. At each reconfigurable system, the System Registry provides access to the reconfiguration-related meta-information and a consistent view of the current status and its established (i.e., installed) capabilities.

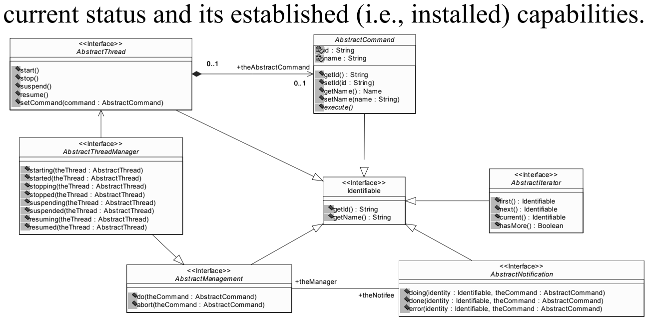

Figure 2. The basic management framework interfaces.

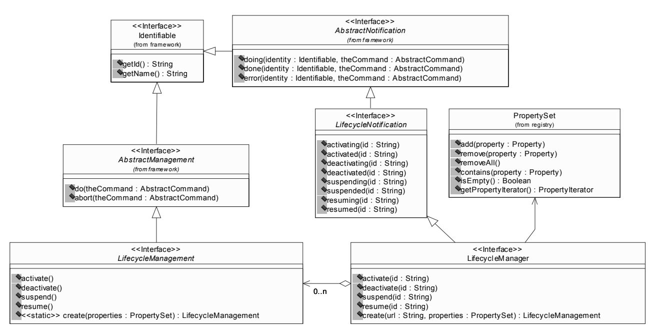

Figure 3. The lifecycle management interfaces.

The *Download Manager* undertakes secure download of protocol implementations from online repositories, handling multiple downloads simultaneously. Once a download is completed successfully, protocol meta-information related to execution environment settings as well as to necessary pre- and post- installation actions is retrieved and used to update the System Registry database.

The *Installation Manager* is responsible for installing and uninstalling implementations of communication protocols and updating the system configuration databases held in the System Registry.

The *Load Manager* undertakes to load object code from installed implementations into the system's main memory and to provide type-safe interfaces of the loaded objects. Depending on the actual implementation class format, this step may not be necessary [8].

The *Lifecycle Manager* coordinates the activation and deactivation of multiple protocol instances belonging in the same protocol layer. It interfaces to operating protocol instances to coordinate protocol state management policies so as to ensure consistency of internal protocol state across reconfigurations as well as persistence of behavior-affecting protocol state across different protocol implementations. In addition, it supports the reconfiguration sequence by acting as proxy for managing the temporal ordering of control commands issued towards its subordinate protocol instances. That includes control commands of other protocol instances and reconfiguration-related commands.

The *Provision Manager* sets proper values to environmental parameters and configuration attributes that each protocol implementation exploits in realizing its functionality. The identification of necessary parameters exploits configuration metadata in the System Registry that is updated by the Installation Manager after a successful installation.

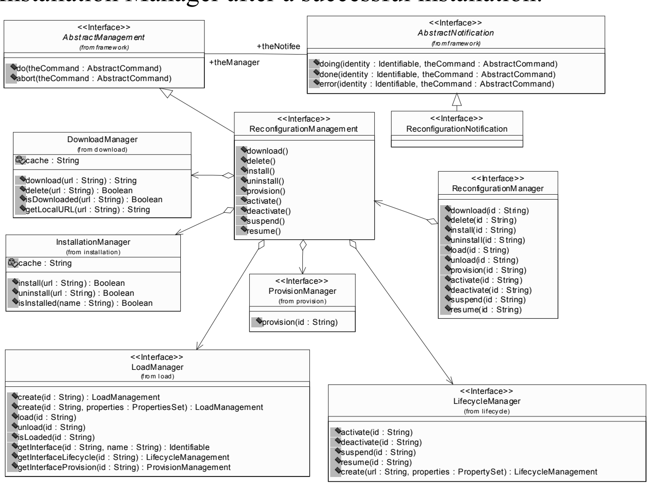

Figure 4. The reconfiguration management interfaces.

Finally, the *Reconfiguration Manager* deals with the reconfiguration of communication standards (as defined by an RDF-encoded specification of functional components and their inter-connections) that may comprise multiple protocol layers. It guarantees the structural consistency of the communication standard and coordinates signaling between protocol instances and the Download, Provisioning and Lifecyle manager objects to achieve reconfiguration.

6. CONCLUSIONS AND FUTURE WORK

In the beyond 3G mobile era, performing and managing reconfiguration actions over mobile systems and networks will constitute a major part of service provision. Due to their constrained resources and limited capabilities, mobile devices will have difficulty in coordinating reconfiguration procedures that may affect multiple network elements or require higher layer context information. Outsourcing some reconfiguration concerns to service provision platforms will ease the load imposed on mobile devices by reconfiguration. We introduced a generic staged procedure and a platform-independent framework for reconfiguration management specified in UML that can be formally mapped to suitable implementation technologies using OMG mappings. We are in the process of validating the effectiveness our framework in a prototypical Java implementation on Linux OS. Future directions of our work foresee integration of our framework with the operating system through appropriate adapters and the export of open (OMD IDL) framework interfaces.

REFERENCES

[1] V. Gazis, N. Houssos, A. Alonistioti, and L. Merakos, "Evolving perspectives of 4G mobile communication systems", Proc. 13th IEEE Personal Indoor Mobile Radio Communications, Lisbon, Portugal, 2002.

[2] Z. Tang, "Dynamic reconfiguration of component-based applications in Java", M.Sc. thesis, MIT, September 2000.

[3] J. Bickle, "Software Radio Architecture (SRA) 2.0 technical overview", OMG TC meeting, December 11, 2000, Orlando, Florida.

[4] OMG SWRADIO SIG document 01-03-03, available from http://www.omg.org/.

[5] RFC 3444, "On the difference between information models and data models".

[6] A. Dey, "Providing architectural support for building context-aware applications", PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.

[7] N. Houssos, V. Gazis, A. Alonistioti, "Application-transparent adaptation in wireless systems beyond 3G", m-Business 2003, 23–24 June 2003, Vienna, Austria, pp. 43-54.

[8] J-Consortium JEFF file format, available from http://www.j-consortium.org/jeffwg/.