RECONFIGURABILITY MANAGEMENT ISSUES FOR THE SUPPORT OF FLEXIBLE SERVICE PROVISION AND RECONFIGURABLE PROTOCOLS

Athanasia Alonistioti (Univ. of Athens, Greece, nancy@di.uoa.gr); Fotis Foukalas (Univ. of Athens, Greece, foukalas@di.uoa.gr); Nikos Houssos (Univ. of Athens, Greece, nhoussos@di.uoa.gr)

ABSTRACT

The expectations for the beyond 3G systems related to service provision will be vested with high flexibility and reconfigurability capabilities in system and protocol level. The management of such complexity becomes an issue of high importance. Efforts undertaken under various projects and standardization activities are targeting the introduction of the optimal mechanisms to cope with this aspect. In this paper we present a framework and mechanisms that provide high potentials for providing solutions for reconfigurability management and support of flexible service provision and reconfigurable protocols.

1. INTRODUCTION

The introduction of 3G networks aims to pave the way for flexible, customized, ubiquitous mobile service provision [1], envisioned in the long term for 4G systems [2]. Contrary to the application specific mobile system design of 2nd generation systems, the potential for flexible and adaptive service offerings is expected to foster the establishment of advanced service provision schemes, engaging business players other than the mobile operator, for example value-added service providers (VASPs) or content providers [1]. This fact, combined with the requirement for anywhere, any terminal, personalised access to services substantially raises the bar for service management, since it should now handle sophisticated inter-domain interactions and cater for application delivery in disparate, reconfigurable environments. Intelligent service provision platforms that mediate between various entities and undertake the bulk of the application deployment and management burden provides for this issue a solution with unique advantages.

Reconfigurability is a key enabling technology for advanced mobile service provision mechanisms. Reconfigurability represents a unique opportunity to provide ubiquitous connectivity to handheld devices over a limitless range of communication standards. However, for reconfigurable systems to realize their full potential they must encompass an advanced reconfigurability management scheme and they can achieve this through the dynamic deployment of reconfigurable components. Within the framework of MOBIVAS and ANWIRE [4], [5], we have introduced architectural approaches and mechanisms for management of reconfigurability for the support of flexible service provision and we investigate the usefulness of component-based software engineering (CBSE) for the implementation of reconfigurable software communication systems. In terms of protocol reconfiguration, our objective is the introduction of a Reconfigurable Protocol Stack through the dynamic deployment of SW components. Moreover, we focus on policy-based reconfiguration management procedures in order to select, negotiate and perform the appropriate reconfiguration actions (i.e. download SW).

Re-usability and management of the complete protocol stack requires that components are largely de-coupled from each other, i.e. they can work without any knowledge about the implementation details of other components they interact with. Component-based software development approach is based on the idea to develop software systems by selecting appropriate off-the-shelf components and then to assemble them with a well-defined software architecture [6], [7]. On the other hand, in order to support flexible service provision in reconfigurable environments, a middleware platform enabling policy based reconfiguration management, service discovery and advanced profile management has been introduced [3]. This work will continue in E2R project, by enhancing the architecture and mechanisms for the support of end-to-end reconfigurability.

In the next sections, we will elaborate on issues related to flexible service provision, reconfiguration management and protocol reconfiguration.

2. RECONFIGURABILITY MANAGEMENT FOR THE SUPPORT OF FLEXIBLE SERVICE PROVISION

Systems beyond 3G will have to revolutionise the business models in order to capitalize their full potential. Introduction of multiple players in the mobile value chain (e.g., network operators, user application developers, content providers, etc) are envisioned that will compete alongside the incumbents for a portion of the revenue stream. The foreseen diversified value chain will require more flexible, reconfigurable and co-operating networks, as well an overall intelligence that will dynamically control the behavior of engaged computing and communication resources. The resulting diversity in business models means that a highly configurable service delivery chain will be necessary [8]:

• *Dynamic user registration*. On-demand user registration to service discovery facilities that are available under the current communication capabilities.

• *Rapid and easy service deployment*. The "walled garden" approach to mobile service provision, with its proprietary instrumentation and long time-to-market will face intense competition in the dynamic mobile value chain, where supporting technologies built on open standards (e.g., IETF protocols, W3C standards, open APIs) will provide a cheap, fast and developer-friendly way of building and deploying value-added services and mobile applications.

• *Reconfigurability management*. Reconfigurability is understood as a major enabler of future mobile systems, while network and terminal reconfigurability management is considered as complicated task that spans multiple layers of the network infrastructure, involves the constant monitoring of network state and requires interaction with underlying equipment that (possibly) resides in multiple administrative domains.

• *Flexible service provision*: Mobile users will value a single entry point, through which the discovery and optimal provision of a plethora of services is performed and which can be tailored to the instant service provision context (e.g., terminal capabilities, user location, network characteristics, etc) as well as their personal preferences.

• *Flexible and user-friendly billing*. Ideally, a single bill should be generated per user for all consumed services, including network and value-added ones. That requires advanced billing systems that collect and process data from various network infrastructures and enable the on-the-fly calculation of pricing schemes, the dynamic modification of tariffs and revenue sharing between the involved parties.

• *Metadata and profiles*: Various profiles should be handled at any time (user, service, network etc.) and managed effectively Thus, profile flexibility, efficiency and control granularity are strong architectural requirements.

• *QoS management*: Users will want to register on demand with available service providers, discover and download services and media-rich content, so QoS provisioning must be supported by the architecture.

The proposed software framework (Figure 1) is able to incorporate all of the above aspects and has been introduced in the MOBIVAS and ANWIRE projects. It is designed to support advanced business models. The entire framework comprises several distributed software components, as shown in Figure 1, whose functionality is briefly described in the following:

- The *Reconfiguration Control and Service Provision Manager (RCSPM)* is the central component of the platform, catering for the automatic service registration and deployment by 3rd party VASPs including the necessary reconfiguration actions on the underlying network for optimal service provision and the management interactions between mobile network/open API providers and VASPs. Moreover, it handles service adaptability to highly variable environments [12].
- The *Charging, Accounting and Billing system (CAB)* [9], calculates the overall user charges induced by service consumption and apportions the aggregate revenue among the involved business parties (e.g., VASPs, service platform providers, network operators). The system has been designed to allow the application of arbitrarily complex billing models and tariffing schemes and to be dynamically reconfigurable by the RCSPM.
- The *VASP* component of the platform is located in the VASP domain (e.g., enterprise servers) and handles secure terminal access to a repository of application clients.
- The *End-User Terminal Software (EUT)* [10], which resides in the end-user terminal (it is not depicted in Figure 1), incorporates an execution environment for applications and reconfiguration functionality. It is worth noting that the EUT needs not be built-in the terminal; it can itself be downloaded and dynamically installed, by any user that can become the appropriate authorization.

An important observation in 1 relates to the fact that the service platform is mainly orthogonal to the actual end-user applications and does not directly interact with them in most cases; that is, service code is not aware of the platform existence. A major advantage of this approach is that platform agnostic (e.g., legacy) applications can be seamlessly accommodated. However, there is an exception to this rule: open interfaces are exposed by the RCSPM and can be invoked by service logic for tasks like charging/billing and user context information retrieval.

As mentioned in the preceding paragraph, the Reconfiguration Control and Service Provision Manager (RCSPM) is the main component of the proposed architecture. It is responsible for a number of important tasks including mechanisms for application deployment and data management, service discovery by end-users, VASP and user authentication and adaptation of service provision to user preferences and terminal characteristics. The architecture of the RCSPM is depicted in Figure 1.

In the following paragraphs, the functionality of the main modules of the RCSPM is presented.

The ***User Interaction Management Module (UIMM)*** is responsible for providing the user with a highly personalizable, context-aware mobile portal [11].

The ***VAS Registrar Module (VASREGM***) is responsible for interacting with 3rd party service providers. Through the VASREGM the platform operator provides VASPs with a way to automatically deploy their services.

The ***Reconfiguration Manager (RCM)*** undertakes network, platform and service reconfigurability. Moreover, it supports the context-aware adaptation of service delivery (e.g., service discovery, service configuration).

The ***Packaging and Downloading Module (PDM)*** is responsible for creating a single downloadable bundle containing all the software components and other supporting resources (e.g., images, etc.) required for executing a service.

The ***Value-Added Service Database Module (VASDBMM)*** is essentially a front-end to a database where service profile information is stored. Service data can be managed (inserted/deleted/updated) by the corresponding VASP.

The ***User Database Management Module (UDBMM)*** handles user profile information required for customized service provision.

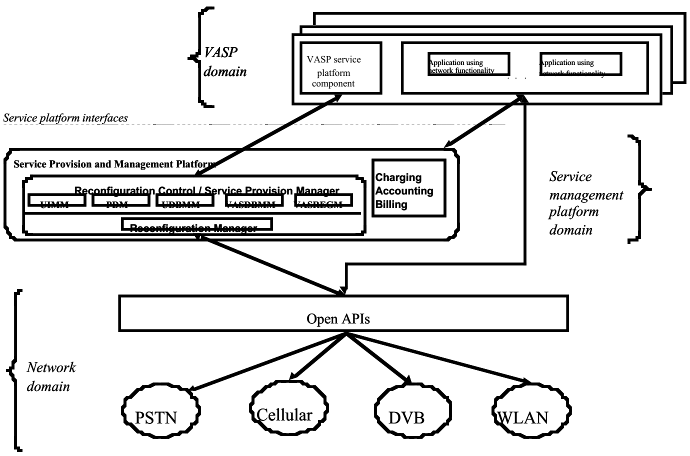

Figure 1 Middleware framework for flexible service provision and reconfiguration management.

2.1. Enabling Technologies

The enabling technologies for reconfigurability and adaptability management in the beyond 3G era that introduce high potentials in future mobile communications can be identified in:

• *Middleware technologies*. The consistent provision of 3rd-party applications in a nomadic context as diverse and technologically heterogeneous as the 4G one, is not a trivial task. Typical application development practices become complicated by the need to anticipate all possible networking contexts during application design and application developers become overwhelmed by having to interpolate network-dependent code in the application's core business logic. It is therefore necessary to provide a middleware layer that abstracts the technological details of the underlying mobile network infrastructure and facilitates a focused application development, whilst also providing open interfaces for a number of important tasks (e.g., user profile management, policy management, QoS management, network abstraction, dynamic user and service registration, etc) [13].

• *Context awareness and adaptation*. Adaptability is commonly defined as the ease with which an entity can be dynamically modified (adapted) according to its context. Adaptation refers to the transition to another state of the entity and may include an alteration in the entity's behavior. Context refers to any information that can be used to characterize the situation of the entity. Despite being a non-crucial factor in the past, adaptation has been widely recognized as a crucial enabler for next generation mobile services, which will be delivered in a variety of circumstances that will not be a-priori predictable. The constantly changing context will considerably impact service access and provision functionality across all functional layers and planes. The corresponding components should therefore be dynamically adaptable to rapidly varying conditions.

• *Frameworks for reconfiguration*. The advent of frameworks such as OSA/Parlay increases the openness of networking infrastructures (e.g., UMTS). OSA/Parlay capabilities located in specialized servers expose selected network features and allow the dynamic re-configuration of network functionality, thereby facilitating a wide range of previously unimaginable applications.

• *Reconfigurability control middleware*. In 3G and post-3G mobile service provision, the role of middleware is not restricted to handling fundamental (albeit important) tasks like seamless "binding" and communication between distributed application objects; it should also incorporate higher-level functionality to facilitate adaptable service provision. Furthermore, deployment and use of diverse distributed applications may require provisioning – or even reconfiguration – of network equipment in multiple administrative domains, an issue best tackled by service provision platforms that can act as a control point for service provision flexibility and adaptability. Systems that are reconfigurable may change both in terms of structure (e.g. by downloading and instantiating a new radio interface and its associated protocol stack) and behaviour (e.g. by reconfiguring the operational parameters of a particular protocol). Reconfigurability management marries the network-level constituents that participate in the service provision with the higher-layer management intelligence required to enforce interoperability and consistency between network services and 3rd-party applications over time and across what may be different implementations of functionality. To this end, the principles of open API frameworks can be exploited to decouple parts of the middleware architecture to achieve abstraction from implementation instruments and to facilitate runtime middleware evolution.

• *Reconfigurable signal-processing algorithms (RSP)*. Reconfigurable signal processing tries to bridge the gap between dedicated ASICs and programmable DSPs by offering a better price/performance ratio through the provision of optimal processing of a wide range of algorithms. RSP merges the programmability of the FPGA approach with an architecture that is fine-tuned to signal-processing tasks. This means that the RSP solution is silicon and computationally efficient, striking a high performance. Nonetheless, major challenges must be overcome when mobile communication components are designed specifically with runtime reconfiguration in mind. Predicting which parts of the processing are most vulnerable to specification changes and would therefore benefit from being reconfigurable is an extremely important consideration that can be both difficult and time-consuming. This is an ongoing concern due to sheer design complexity, but using RSP to deal with specific functions minimizes the overall design risk. Where reconfiguration is done on the fly in order to share hardware across multiple functions, the scheduling of functionality switching needs to be carefully considered.

• *H/W and S/W co-design methodologies*. According to the IEEE DASC Co-design Study Group, "HW/SW co-design is a design methodology supporting the concurrent development of hardware and software (co-specification, co-development and co-verification) in order to achieve shared functionality and performance goals for a combined system". Co-design comprises co-specification, system architecture definition, partitioning, synthesis and co-verification. The purpose of co-specification is to capture the requirements of the complete system. The implementation architecture and style of the system is defined at architecture definition phase. The purpose of partitioning phase is to divide the functionality of the complete system into hardware and software, which are designed in technology dependent synthesis phase. Co-verification is needed in each phase to ensure that transformations are done correctly and that the intended functionality is preserved throughout all the phases.

• *Component-Based Software Engineering (CBSE):* is a_ approach for the development of technologies that enable component-based plug-and-play style of systems development. Among the enabling technologies, component generation or manufacturing tools are becoming mature. However, technologies for integrating or assembling components are still lagging behind.

3. RECONFIGURABLE PROTOCOLS

Different approaches for the reconfigurable terminal/network node architectures coexist according to the technologies that can be supported. An important aspect of reconfigurability implementation is the introduction of architectures and mechanisms for a reconfigurable protocol stack.

The main objective is the definition of reconfigurable programming interfaces for both radio and networking layers of the architecture model in order to gain access to the interfaces functionality of standard communication protocols. Reconfigurable programming interfaces will be used to expose the functionality of functions performing reconfiguration. These programming interfaces can be defined using a formal language (Java) or framework (CORBA) which supports an object-oriented approach. In view of the highly demanding requirements for such architectures and mechanisms, component-based software engineering (CBSE) may provide a powerful option. In contrary to an object, a component may provide a richer range of intercommunication mechanisms, a higher degree of re-use and adaptability, and usually a larger granularity. A component may comprise one or more objects [13]. It is apparent that component-based software development techniques appear with high programmability and flexibility in protocol design and specification.

Granularity and interface design are the most important issues to achieve flexibility. Moreover, re-usability requires that components are largely de-coupled from each other, i.e. they can work without any knowledge about the implementation details of other components they interact with. Component-based software development approach is based on the idea to develop software systems by selecting appropriate off-the-shelf components and then to assemble them with a well-defined software architecture.

A component has three main features:

1. a component is an independent and replaceable part of a system that fulfils a function;
2. a component works in the context of a well defined architecture; and
3. a component communicates with other components by its interfaces .

Towards a component-based architecture approach the main steps encompass the protocol description in component-based terms that are consistent with Component-based Models. Protocols implemented in Java running inside the Java Virtual Machine (JVM) [14] on top of an operating system (OS) suffer from a performance penalty compared to protocols implemented in the OS kernel [15]. However, for rapid testing and monitoring it is more important that protocols can be modified, extended and configured easily [3].

The life cycle of component-based architecture for reconfigurable protocols can be summarized as follows:

1) Requirements analysis for protocol stack reconfiguration [14]
2) Software architecture selection, construction, analysis, and evaluation (Corba Component Model - CCM, Enterprise Java Beans - EJB)
3) Component identification and customization
4) Protocol integration
5) Protocol testing
6) Software maintenance [16]

Also, an important aspect is the development of appropriate intercommunication mechanisms in order to achieve component integration. Component integration introduces two issues: 1. packaging components so that they can be connected at runtime, and 2. connecting, disconnecting, and re-connecting components at run-time[16].

Several steps for a component-based protocol specification can be identified. We identify these steps for the PDCP UMTS protocol [17] paradigm:

1. Identification of PDCP protocol functions
2. Separation of functions used for communication between two protocols/layers (intra-functions), i.e. PDCP and IP, and functions that describe the functionality within PDCP protocol (inter-function)
3. Mapping of protocol intra-functions to components. Components encapsulate functions of implementation

and expose a set of interfaces (information) in order to manage the protocol functionality
  - Description of functionality with independent components
  - Demarcation of functionality into sub-function common in other protocols
4. An overall description of the protocol in order to ensure a uni-directional reliable transfer of data.

It is possible that for reconfigurable mobile terminals to realize their full potential, they may have to achieve reconfigurability through the dynamic deployment of software components. A dynamic deployment of software components needs an interoperable environment for different protocol components, particularly a component-based framework for software communication protocols that will have to be defined. Such an environment aims to define a middleware that allows protocols and its parts to interoperate over different hardware mobile terminal designs. A framework for dynamic deployment of hardware components have been introduced (JTRS Software Communication Architecture - SCA) that maximizes the independence of software from hardware [18]. The management framework will comprise interface and behavioral specifications, general rules and security requirements to meet open reconfigurability management architectures.

Moreover, in order to develop a reconfigurable protocol stack, the different layers that compose the system would also have to be modified to allow for layer reconfiguration. This has been tackled by introducing the concept of a Reconfiguration Management Plane affecting each layer [19]. An illustrative architecture of the proposed reconfigurable protocol stack is given in Figure 2. The key element of this architecture is the Reconfiguration Management Plane and the interaction with the generic policy-based reconfiguration manager. Important aspects of the proposed approach is the introduction and enhancement of the architecture of dynamic protocol reconfiguration through software downloading and offloading mechanisms. These mechanisms will complement the object oriented and component based approaches to protocol reconfiguration and will provide a powerful tool for reconfiguration and resource management.

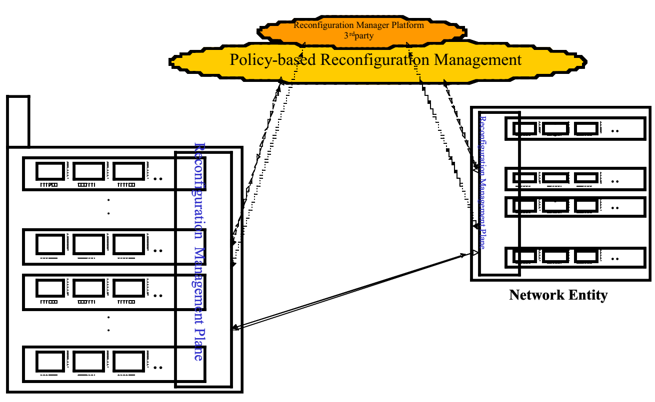

Figure 2. Framework for protocol reconfiguration

4. CONCLUSIONS

Reconfigurability management is an important aspect for the introduction of advanced service provision features in the communication systems beyond 3G. The introduction of advanced functionality in order to support flexible service provision and protocol reconfiguration will be one of the main research targets in mobile computing and wireless communications. Parts of the overall problem have been tackled under various research and standardization activities. An integrated approach becomes necessary in order to enable effective and holistic solutions to take effect. In this paper we have summarized some main research directions identified in the context of MOBIVAS and ANWIRE projects that will be enhanced and integrated in End to end Reconfigurability efforts.

10. REFERENCES

[1] UMTS Forum Report No. 9, "The UMTS third generation market - structuring the service revenues opportunities", available from http://www.umts-forum.org/.

[2] J. Pereira, "Beyond third generation", Wireless Personal Mobile Communications (WPMC), 22 September 1999, Amsterdam, The Netherlands.

[3] N. Houssos, V. Gazis, A. Alonistioti, "Enabling delivery of mobile services over heterogeneous converged infrastructures", Special Issue of the Kluwer Information System Frontiers Journal on "Network convergence: Realizing a Dream or a Nightmare?", accepted for publication.

[4] http://mobivas.cnl.di.uoa.gr/

[5] http://anwire.org/

[6] X. Cai, M. R. Lyu, K.-F. Wong: Component-Based Embedded Software Engineering: Development Framework, Quality Assurance and a Generic Assessment Environment. International Journal of Software Engineering and Knowledge Engineering 12 (2): 107-133 (2002)

[7] S. Kim, J. Masse, S. Hong. " Dynamic Deployment of Software Defined Radio Components for Mobile Wireless Internet Applications" Second International Human.Society@Internet Conference (HSI). pp. 694-700, Seoul, Korea, June 2003

[8] M. Dillinger, K. Madani, N. Alonistioti, "Software Defined Radio: Architectures, Systems and Functions", John Wiley and Sons, Ltd ISDN: 0-470-85164-3, 2003

[9] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, "Charging, Accounting and Billing as a Sophisticated and Reconfigurable Discrete Service for next Generation Mobile Networks", Fall VTC2002, Vancouver, Canada, September 2002

[10] O. Fouial. K. A. Fadel, I. Demeure, “Adaptive Service Provision in Mobile Computing Environments”, 4th IEEE International Conference on Mobile Wireless Communication Networks (MWCN 2002), Stockholm, Sweden, 9-11 September 2002.

[11] N. Houssos, S. Pantazis, A. Alonistioti, “Towards adaptability in 3G service provision”, IST Mobile Communications Summit 2002, Thessaloniki, 16-19 June 2002.

[12] N. Houssos, S. Pantazis, A. Alonistioti, “Generic adaptation mechanism for the support of context-aware service provision in 3G networks”, 4th IEEE Conference on Mobile and Wireless Communication Networks (MWCN 2002), Stockholm, Sweden, 9-11 September 2002.

[13] C. Mascolo, L. Capra, W. Emmerich, “Mobile Computing Middleware”, Networking 2002 Tutorials, LNCS 2497, pp.20 – 58, 2002.

[14] M. Jung, E. Biersack, A. Pilger, “Implementing Network Protocols in Java-A Framework for Rapid Prototyping”, ICEIS 1999: 649-656

[15] N.C. Hutchinson, L.L. Peterson, “The x-Kernel: An Architecture for Implementing Network Protocols” IEEE Transactions on Software Engineering 17 (1) 1991

[16] W. Kozaczynski, J.Q. Ning, “Component-Based Software Engineering (CBSE)”, 4th International Conference on Software Reuse, 1996

[17] 3GPP TS 25.323: “Packet Data Convergence Protocol (PDCP) Specification”.

[18] http://www.sdrforum.org/tech_comm/mobile_wg.html

[19] M. Barbeau, F. Bordeleau, "A Protocol Stack Development Tool Using Generative Programming," in: D. Batory and C. Consel (Eds.), Proceedings of Generative Programming and Component Engineering (GPCE), 2002