Prototyping Environment for Equipment Reconfiguration Management and Control

A. Katidiotis, V. Stavroulaki, P. Demestichas
University of Piraeus
Greece
katidiot@unipi.gr

M. Muck
Motorola Labs
91193 Gif-sur-Yvette, France
Markus.Muck@Motorola.com

B. Steinke, R.K. Atukula, U. Lücking
Nokia Research Center
Bochum, Germany
Bernd.Steinke@nokia.com

E. Patouni, P. Magdalinos
University of Athens
Greece
elenip@di.uoa.gr

***Abstract*— The course of events in the wireless communications area has driven the research towards the Beyond Third Generation (B3G) systems. The B3G concept leans on the co-existence and co-operation of multiple, dissimilar Radio Access Technologies (RATs). The Reconfigurability concept has been developed to accommodate and exploit the advantages offered by B3G systems. Reconfigurability supports the B3G concept by providing equipment (terminals and network elements) with capabilities to dynamically select and adapt to the most appropriate RAT. In order for this to be realised, an enhanced equipment management and control functionality should be introduced. Such architecture is presented on this paper that focuses on the prototyping environment and the functional entities of the equipment reconfiguration management and control architecture. Also, the correlation between the concept of Autonomic Computing and this architecture is presented.**

***Index Terms*— Equipment control, Equipment management, Reconfigurability**

I. INTRODUCTION

THE evolution of wireless communications over the last years has resulted in a clear trend towards Beyond Third Generation (B3G) systems, which support the integration and co-existence of multiple, diverse Radio Access Technologies (RATs) in a common composite radio environment. A typical composite radio scenario may include access technologies, such as GPRS, UMTS, WLAN IEEE 802.11, WiMAX IEEE 802.16 and DVB. The exploitation of B3G systems can be facilitated by the Reconfigurability concept, which is an evolution of "software defined radio" [1]. The application of the Reconfigurability concept in an environment with a multitude of wireless access technologies provides mechanisms for enabling equipment (terminals and network elements) to dynamically select and adapt to the most appropriate RATs, in a seamless, transparent and robust (safe, secure, reliable) manner, according to specific service area region conditions and time zones of the day. Reconfigurability calls for enhanced equipment management and control functionality. The contribution of this paper falls within the development of a *Prototyping Environment* for *Equipment Reconfiguration Management and Control (ERMC)* for enabling the realization of the seamless experience, carried out in the E²R Project [2][3]. This paper is structured as follows: The second section presents a *ERMC* architecture while the following two sections discuss the functionality and structure of each of its modules. For each of the main modules of the *ERMC* architecture a preliminary prototype has been developed. The fifth section presents an overview of the potential integrated prototype. The sixth section outlines autonomic aspects of the presented framework. Finally, some standardisation aspects are presented in the seventh section. The paper concludes with a short summary of the major concepts presented.

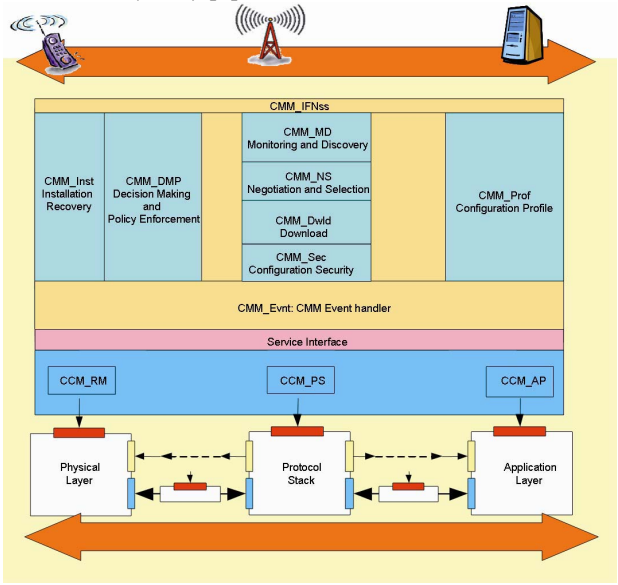
II. ARCHITECTURE

The configuration control framework for the reconfigurable equipment architecture consists predominantly of three types of modules [4]:

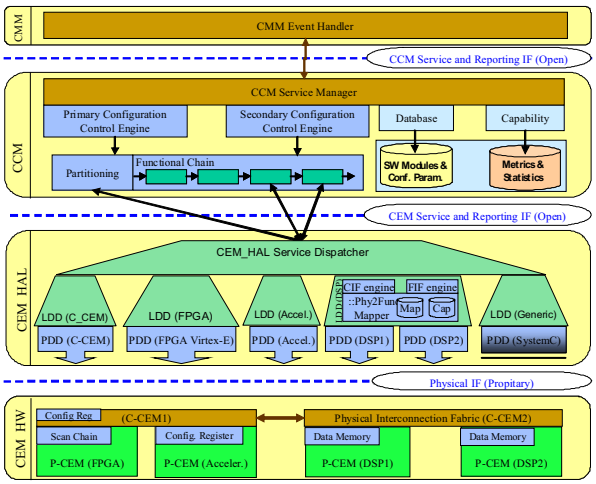
- ***Configuration Management Module (CMM)***. The CMM which manages the reconfiguration processes according to specified semantics, protocols and configuration data model, and is responsible for deciding on the correct functional configuration for the equipment.
- ***Configuration Control Module (CCM)***. The CCM is responsible for interpreting configuration requests from the CMM into actual implementations on the hardware.
- ***Configurable Execution Module (CEM)***. A CEM is a configurable hardware resource. One or more CEMs are configured by the CCM.

These main modules are further split into several sub-modules that cover certain equipment specific tasks, as there are software download and installation, service monitoring, negotiation and selection, etc. A logical overview of the equipment management and control architecture, in terms of

involved sub-modules is depicted in Figure 1 (a). Figure 1 (b) shows in more detail the relationship between the different CCM and CEM modules. The CEMs are the lowest (physical) layer and represent the signal processing CEM (P-CEM) and Communication CEM (C-CEM) hardware. To enable the actual CEM hardware abstraction through the CEM_HAL, three sub layers are introduced: The CEM_HAL Service Dispatcher, the Logical Device Driver (LDD) and the Physical Device Driver (PDD) [5].



(a)



(b)

Figure 1. Overview of Equipment Management & Control Architecture: (a) CMM and CCM (b) CCM and CEM

III. CONFIGURATION MANAGEMENT MODULE

The CMM is a functional entity within the equipment that is responsible for the management of all configuration tasks in the equipment as well as the negotiation of reconfiguration decisions with other entities. This way it manages the distributed controllers, which will initiate, aggregate and coordinate the different reconfiguration functions. Moreover, the CMM should interact through its external interface with other equipments of different types, located either in the equipment or network domains. The CMM [6][7] comprises the following modules/entities:

The "***Network Support Services Interface***" (***CMM_IfNss***) module is considered as the main interface that is provided by all the CMM modules to the network encompassing the equipment. This entity is responsible for the bi-directional reconfiguration related communication between the network support services and the CMM modules. It mediates between the requests of the CMM modules towards the network and dispatches the responses from the network to the corresponding modules.

The "***Configuration Profiles***" (***CMM_Prof***) module aims at providing configuration profiles information on applications, user classes, equipment classes/capabilities and configuration data models. It manages the profile repository and retrieves profiles on request from other modules.

The "***Monitoring and Discovery***" (***CMM_MD***) module identifies the available networks in a certain area and monitors their status. This entity can conduct more complex assessments, taking into account data from multiple CCMs. It acquires information on the context in the environment of the device. In case any discrepancies are observed it may be rendered necessary to initiate re-selection of the most suitable radio access scheme.

The "***Negotiation and Selection***" (***CMM_NS***) module is targeted to the negotiation of offers with the various available networks. The selection procedure takes into account information such as the user and equipment profile and the offers negotiated with the networks. The network offers specify the services offered, the QoS levels supported and cost related information.

The "***Configuration Downloads***" (***CMM_Dwnld***) module provides the capability to perform downloads of the different components for the reconfiguration process. It undertakes the management of the downloading procedure.

The "***Configuration Security***" (***CMM_Sec***) module is responsible for security functions required during the reconfiguration process within the different layers.

The "***Decision-making and Policy Enforcement***" (***CMM_DMP***) module is a module that mainly supports context and policy management procedures. Policy-based mechanisms and procedures are being implemented and performed by entities that are dedicated to mode/standard selection.

The "***Reconfiguration Installation***" (***CMM_Inst***) module provides the means for configuration representation and configuration deployment (validation, installation, switch).

The "***Configuration Event Handler***" (***CMM_Evnt***) module provides a reporting interface to the CCM. Via this interface the CMM_Evnt receives messages and trigger events from the CCMs and dispatches them to the appropriate sub-module of the CMM for processing. All communication, independent of the direction, between CMM sub-modules and the different CCMs is processed via the CMM_Evnt.

IV. CONFIGURATION CONTROL MODULE

The Configuration Control Module (CCM) initiates, coordinates and performs the different reconfiguration functions within the respective reconfigurable subsystems of the equipment. This section describes in detail the interfaces provided by the CCM.

A. *Primary Configuration Control Interface*

This is the primary interface between the CMM and the CCM which sets-up a new configuration; it translates the configuration in to a set of configurations for the different layers (application, protocol and physical layer). The configuration requests could be at the level of a complete RAT but could also be at a higher granularity. In a high granularity description sub components of the radio modem or protocol stack would be defined as well as the interconnection between them. Typically the functional components could be channel or speech codec's.

B. *Secondary Configuration Control Interface*

The secondary interface between the CMM and the CCM supplies a set of functions unique to the selected RAT. These functions might initiate a change request from the RRC/Radio Link Controller (RLC) in layer 3 of a WCDMA protocol stack and would, for example, request a change in the number of transport channels. Precisely how this functionality is changed is implementation dependant but could involve actually changing the system configuration (i.e. changing allocation of resources) or by modification of parameters within the system.

C. *Capability Interface*

The Capability Interface allows the CMM to determine what configurations the underlying layer is able to support.

D. *Database Interface*

The Database Interface provides access to databases in the different layers. The database will contain layer specific configurations and their properties. It may also contain metrics that can be used by the configuration plane to determine future configurations.

V. INTEGRATED PROTOTYPE

Preliminary distinct prototypes corresponding to the CMM, CCM and CEM modules have been implemented in order to obtain a first proof of concept of the architecture presented [8]. These prototypes will be further elaborated on in the context of the work performed E^{2}R II [3]. The prototypes will be used for further development and evaluation of necessary improvements and will be extended with physical layer reconfiguration capabilities and network support services. Moreover, these individual prototypes will be integrated into a unified prototyping environment for reconfigurable equipment.

Equipment prototyping focuses on two aspects: terminal and network element, i.e. base station prototyping. Therefore, two proof of concept architectures will be developed:

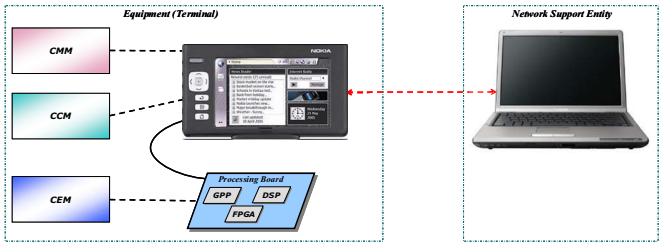


Figure 2. Reconfigurable terminal prototype architecture

Figure 2 depicts the prototype proof of concept architecture for the reconfigurable terminal. This consists of a NOKIA 770 Internet tablet [9] and a processing board, which comprises a General Purpose Processor (GPP), a Digital Signal Processor (DSP) and a Field-Programmable Gate Array (FPGA). It is envisaged that the processing board and the NOKIA 770 will be connected through a USB link. Regarding the distribution of the functional software entities of the equipment management and control architecture to the hardware architecture depicted in Figure 2, the CMM and CCM modules will located in the NOKIA 770, while the CEM modules will be located in the processing board. It should be noted that the CMM and CCM module implementation will be based mainly on C/C++ maemo [10], a development platform for the creation of applications for the NOKIA 770 internet tablet.

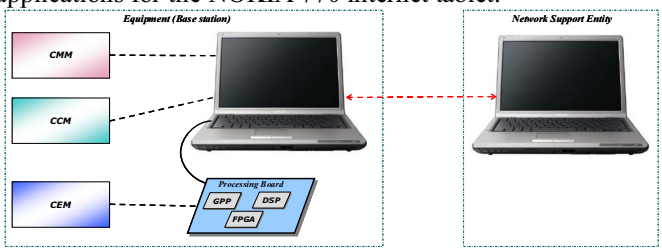


Figure 3. Reconfigurable base station prototype architecture

Figure 3 depicts the prototype proof of concept architecture for the reconfigurable base station. This is similar to the architecture for the terminal with the difference that a laptop is used in place of the NOKIA device. The distribution of software entities to physical entities is also similar as for the previous case, i.e. the CMM and CCM modules will be located in the base station side laptop, while the CEM modules will be located in the processing board.

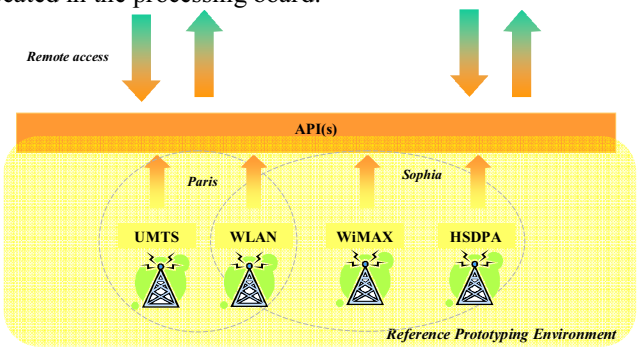


Figure 4. Network infrastructure of Reference Prototyping Environment

Furthermore, these equipment prototypes, both terminal and base station, will be integrated in the E2R Reference

Prototyping Environment, that also comprises a prototype network infrastructure consisting of UMTS, WLAN (IEEE 802.11x), WiMax and HSDPA segments (Figure 4).

The network infrastructure is remotely accessible to the developed equipment prototypes through appropriate Application Programming Interfaces (APIs). This demonstration will be based around a dedicated proof-of-concept environment used to validate the work in the areas of cognitive networks, reconfigurable terminals, enhanced radio resource and spectrum efficiency, dynamic and robust (stable, secure, reliable) reconfigurations. The developments and experiments will also provide numerous outputs and feedbacks for the other workpackages.

VI. AUTONOMIC ASPECTS

The key component of autonomic computing systems is the ability of self-management [11]. The administrator of the system defines some high-level objectives, and the autonomic systems are able to manage themselves. So, for example, it is possible for an autonomic system to monitor its own use, to detect a change in its environment, to find, download, validate and install updated system components. The presented architectures and the relative prototypes need to be enhanced so as be able to operate in accordance with the autonomic computing paradigm.

In more detail, the Equipment Reconfiguration Management and Control architecture should be enhanced so as to provide the intelligence for sensing and analysing what goes on in the service area of the reconfigurable equipment. In other words it should provide support for self-configuration/optimisation/healing functions, e.g., by identifying impairments, performance degradations, failures. Furthermore, the presented architecture should be refined so as to accurately describe the policies and profiles elements that will provide the requirements and the constraints for the self-configuration and self-optimisation functions. Additionally, the Monitoring and Discovery, the Negotiation and Selection and the Profiles modules of the reconfigurable equipment prototype can be enhanced with learning capabilities. Finally, the enhancement and further development of appropriate policies, strategies and algorithms for finding reconfigurations is required.

A. Policy Management

Regarding policies, policy management and access decisions [12] play an important role in management of reconfigurable devices remotely. When a device management server makes a resource request upon a reconfigurable device, an entity called Policy Enforcement Point (PEP) is charged with access control by enforcing authorization. In order to enforce policy, this entity formalizes attributes describing the requester at the Policy Information Point and delegates the authorization decision to the Policy Decision Point. Applicable policies are located in a policy database and evaluated at the Policy Decision Point, which then returns the authorization decision. Using this information, the Policy Enforcement Point in the device can deliver the appropriate response to the management server.

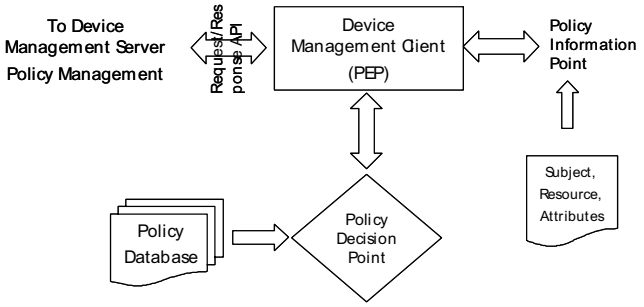


Figure 5. Policy Management

B. Defining Policies

The policies are built in a reconfigurable device which are defined e.g., by manufacturer, regulator, etc. Either user or the administrative server gets control over device management. The key element is the policy object which aggregates other policy objects elements or Policy elements. The Policy element is composed principally of Target, Rule and Obligation elements and is evaluated at the Policy Decision Point to yield an access decision.

As multiple policies may be found applicable to an access decision, (and since a single policy can contain multiple Rules) Combining Algorithms are used to reconcile multiple outcomes into a single decision.

The Target element is used to associate a requested resource to an applicable Policy. It contains conditions that the requesting Subject, Resource, or Action must meet for a policy object, Policy, or Rule to be applicable to the resource. The Target includes a build-in scheme for efficient indexing/lookup of Policies.

Rules provide the conditions which test the relevant attributes within a Policy. Any number of Rule elements may be used each of which generates a true or false outcome. Combining these outcomes yields a single decision for the Policy, which may be "Permit", "Deny", "Indeterminate", or a "NotApplicable" decision.

Policy Attributes provide the typed values that represent both a resource requester and the Policy's condition predicates.

VII. STANDARDIZATION

The upper architectural and algorithmic approaches are prepared to be presented at standardization bodies in the framework of the European Project E^2R Phase II [3]. Among others, suitable bodies are: the Open Mobile Alliance (OMA), the Object Management Group (OMG), the TeleManagement Forum (TMF), the SDR Forum, IEEE P1900, etc.

VIII. CONCLUSIONS

The contribution of this paper falls within the development of a Prototyping Environment for Equipment Reconfiguration Management and Control (ERMC) for enabling the realization

of the seamless experience, carried out in the E²R Project. This paper started with the presentation of such an Equipment Reconfiguration Management and Control architecture and described in detail the functionality of its modules. A preliminary prototype for each of the main modules has been developed. These distinct prototypes will be integrated. The paper presented an overview of the potential integrated prototype and discussed on the corresponding hardware architecture. Finally, the paper ended with the outline of autonomic aspects of the presented framework and some standardisation aspects.

ACKNOWLEDGMENT

This work has been performed in the framework of the EU funded projects End-to-End Reconfigurability Phase I (E²R) and II (E²R II). The authors would like to acknowledge the contributions of their colleagues from the E²R and E²R II consortia.

REFERENCES

[1] J. Mitola, "Software Radio Architecture", Wiley-Interscience, 2000
[2] End-to-End Reconfigurability (E²R) project www.e2r.motlabs.com
[3] End to End Reconfigurability - Phase II (E2R2), http://www.e2r2.motlabs.com
[4] C. Dolwin, S. Mende, J. Brakensiek "The Role of the Configuration Control Module in an End to End Reconfigurable System," Software Defined Radio Technical Conference, November 2004
[5] Bernd Steinke et al, "Hardware Abstraction Architecture based on Configurable Execution Modules for Functional System Chains", WG6 Reconfigurability, WWRF#13 Meeting, Jeju Island, Korea, March 2005
[6] J. Vogler et al, E2R White Paper, "Equipment Management and Control Architecture", July 2005
[7] V. Stavroulaki et al, "A Management and Control Architecture for Enabling End-to-End Reconfigurable Equipment Operation", Wireless World Research Forum (WWRF) 13th Meeting, Jeju, South Korea, March 2005
[8] A. Katidiotis et al, "Prototyping for End-to-End Reconfigurable Equipment", in Proc. of 15th IST Mobile and Wireless Communications Summit, June 2006
[9] http://www.europe.nokia.com/
[10] The Maemo development platform, http://maemo.org/
[11] J. Kephart, D. Chess, "The vision of autonomic computing", IEEE Computer, Vol. 36, No.1, pp. 41-50, January 2003
[12] Trent Jaeger et al. "Policy management using access control spaces", ACM Transactions on Information and System Security (TISSEC) Volume 6 , Issue 3, August 2003