Managing Multi-Configurable Terminals

Radu State, Karim El-Khazen, Georges Martinez, Guillaume Vivier

Motorola Labs, 91193 Gif-Sur-Yvette, FRANCE
{state, elkhazen, martinez, vivier}@crm.mot.com

***Abstract* - The advent of software defined architecture allowing flexible software configuration of terminals demands an increased capability for the management applications in charge with configuring such equipment. We address in this paper an architecture for the configuration of software defined radio, which allows the configuration of multi-technology terminals.**

Keywords: Configurable Terminal, DEN, SDR, management.

1. Introduction

Future wireless networks will provide added value by allowing the deployment of a large variety of services over them. We are witnessing a shift from a technology and transport-oriented view towards a service-oriented one. These services are provided over a heterogeneous infrastructure (UMTS for the radio part, IP for the core network, WLAN, DVB). Recent advances in research on software-oriented radio receivers, which will offer an enhanced flexibility to these devices, determine us to investigate how configuration management for such radios should be provided. Until now, the radio equipments were vertically closed; most of their configuration is done on well established and agreed upon parameters.

Future equipments will provide more flexibility by implementing in software most functions that are performed now in hardware. We propose in this paper the management of such radio receivers based on currently IETF proposed management paradigms. First, we consider the use of information models capturing the essential of distribution, deployment and installation aspects of application. Next, we show its usage within a Java-based management architecture.

Our paper is structured as follows: we will describe the context of our approach in section 2. A brief survey of related research work is given in section 3. Section 4 illustrates our approach in modeling applications for software-defined radios and the management platform. Section 5 concludes the paper with pointers to future work and conclusions.

2. Problem Statement

If a software-defined radio must be reconfigured in order to reflect a new usage profile, as shown in Fig. 1, several problems arise. First, who is reconfiguring the terminal? That is, will the end user demand it (an option which is not viable if we consider the average user), the terminal itself based on evaluation of radio conditions and/or service related aspects, or an external management platform?

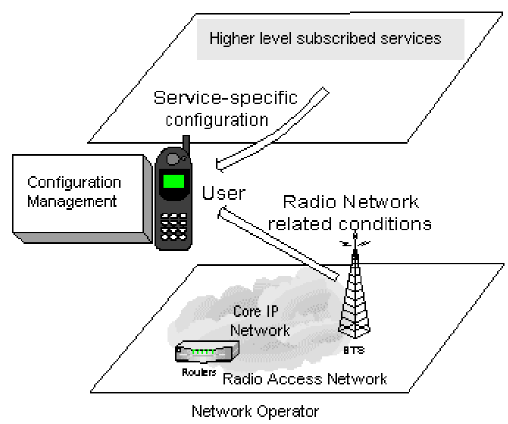

Fig. 1. Management framework.

We consider that for the short term, the later will be the case. In this case, if configuration of software-related entities must be performed, then we must provide on one hand an open management protocol for the management, but also a common information model for the applications and their configuration. This is similar to what made IP management successful, where an easy and common protocol (SNMP) and its associated information model (MIB2) allowed multi-vendor and varied equipments to be managed by third-party management platform. This is particularly interesting in wireless networks, where radio network operators could manage user terminals not necessary belonging to their own networks. Opening the management of terminals towards non-proprietary management platforms is a necessary condition for providing seamless and ubiquitous "Beyond 3G" services anywhere. We propose in this paper to extend the Common Information Model (CIM), an initiative for integrated management of networks, systems and services towards the management of radio terminals. We will show next, how some particular issues must be addressed in order to deploy the management on radio terminals.

3. Related Work

Recent work addresses the issue of providing a larger flexibility of the radio-related entities by defining a standard architecture for the radio equipment. This is proposed in the larger context of software definable radio, that is, by pushing functionalities from hardware-implemented to software and upgradeable components. Some of these approaches propose the definition of a standard architecture based on a set of objects communicating over a CORBA bus [6, 7]. The main goal is to allow the easy reconfiguration of radio equipment by a third party (not necessarily the manufacturer).

This work assumes that a generic architecture for implementing radio equipment can be established. A different solution to flexible configuration is through a management plane, where management actions are issued by a manager, deployed to the radio equipment, and next translated to equipment specific requests. For such a scenario to succeed, the management interface should be open and capable to capture the needed configuration information. An open interface means a standardized communication protocol (like SNMP for IP networks for instance) and by capturing the needed configuration information, we need a generic information model capable to represent different types of equipment.

Thus, management platform should be able to manage multi-vendor equipment in a transparent way. A first approach towards integrated management of the radio terminal has been proposed in [12] as an over-the-air management of a radio terminal. This approach proposes the use of WAP (Wireless Application Protocol) as a transport layer for management information for a particular terminal technology. In [9], a reconfiguration management solution is proposed for a SDR (Software defined radio). The architecture aims to provide the reconfigurability of the software within such advanced equipment.

Attempts to provide a layered architecture for radio equipment were proposed in [11] while the same authors identified constraints for a dynamical physical layer in [10]. An object-oriented approach to model the physical layer has been introduced in [8] in order to represent the resources and functions at this level. The proposed model is mostly an inheritance hierarchy failing to represent important relations among the entities. However, it provides an interesting approach to control this layer.

4. Proposed Solution

The solution comprises two parts: the first one is a model to represent the information needed for the management. The second one is a management architecture that uses the model in order to perform service and network management.

4.1. *Information Modeling*

The information model is built based on the CIM (Common Information Model)/DEN (Directory Enabled Networks) [1,2, 3,4] initiative proposed by the DMTF (Distributed Management Task Force). The CIM/DEN model allows representing management information for IP networks, system and application management. CIM is an object-oriented approach towards modeling and defining managed resources. The latter can be refined though inheritance mechanisms. CIM was developed in order to provide a comprehensive information model to capture the modeling of all entities existing in a network.

Therefore, it captures notions like user, service, service access point, network element, application, providing a framework with more than 700 classes to the management plane. Although, CIM is not implementation-oriented, it is considered that management information modeled with CIM is mapped to the LDAP (Lightweight Directory Access Protocol) [13] format and stored in a central repository. This repository is actually a directory providing a centralized store for information, linking users, services and applications.

The goal is to enable management application a standard representation of the management information for the overall service and thus allow the shift from proprietary network management information stores towards open and easily accessible information sources. On type of management information that is concerned with relationships among existing network elements, users and applications can be regrouped under the general context of policy-based management information. The main idea in policy-based management is to deploy more intelligence onto the network entities and delegate management task to their responsibility.

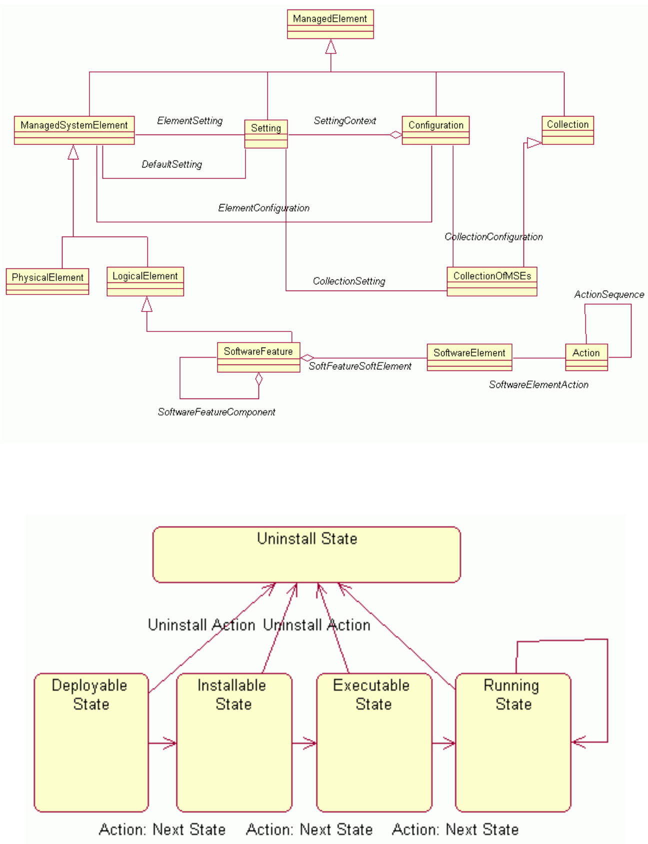

Fig. 2. Information modeling.

Modeling applications and their management is the central activity of the DMTF Application working group. The Application extension model of the CIM (see Fig. 2 for a small abstract) introduces several classes and relationships to be used for modeling software managements. It allows to express notions like software entities (the class Software Element), functional regrouping of entities (the class Software Feature) in order to provide a well-defined function, and global application (the class Application System) providing a particular business function. Software installation and configuration is addressed by proposing a life-cycle model for the class Software Element. It allows modeling the potential states (Deployable, Installable, Executable, and Running) of an application as well as the different actions (installing a feature, starting, configuring, and monitoring).

4.2. *Management Architecture*

The above-mentioned model is used by management architecture as shown in Fig. 3. It is responsible to implement the model, and perform the management tasks. The architecture is composed of different parts.

We illustrate only those components concerned with the application configuration. A central application server stores the different applications. The instantiation is done with respect to the central information model. These applications are stored in an LDAP compliant server. The mapping of the policies to LDAP is done as proposed by the DMTF forum. This server captures also dependencies existing among different applications, radio and terminal based configuration and user profiles.

The Application deployment module must perform three major tasks. Firstly, indicate the terminal that reconfiguration must be done. Secondly, upload the applications to the terminal, and thirdly install the applications and run them.

The first task is done using the simple IETF proposed Script MIB [17]. Simply stated, the script MIB provides the necessary support to distribute scripts for management tasks. It is very useful since executing scripts on remote devices reduces the process load on the management platform. The MIB itself is relatively simple, thus use a modular SNMP agent in order to minimize the computational burden on the wireless device. This simple SNMP agent implementing the MIB is accessed by the management application in order to start the process. These scripts can be executed immediately, or at specific time moments. The later can be done by the use of the IETF proposed Schedule MIB [16].

The Schedule MIB allows performing management actions at periodic intervals or at scheduled times. One more time, the MIB is relatively simple (one scalar and one table), thus a simple subagent implementing this MIB can be added to the platform. Thus, reconfiguration actions can be stored in these MIBs, allowing initiating for instance reconfiguration and management actions on periodic basis or off-hour times. For instance, long lasting application downloads, or gathering application monitoring information can be done at late evening hours. The second task is related to the terminal initiating the software download phase. In this phase, the terminal will contact the LDAP server and request the entries containing the code that must be installed. The LDAP server has entries where serialized Java code or applications binaries are stored to be deployed. The access from the mobile to the server can be done in two ways. If the terminal can connect over TCP directly then it uses the JNDI (Java Naming and Directory Interface) [18] to access the server via a Java stack. Once the request is made, the LDAP server returns the entries. If this is not possible, requests will go through a DSML [14] (Directory Services Markup Language Gateway), in order to map LDAP request to XML [15] encapsulated data. This is currently proposed as a way to allow XML programmers the access to LDAP-enabled directories without having to write to the LDAP interface or use proprietary directory-access APIs. Additionally, it is useful when application provisioning is done with several LDAP servers using potential different schemes. The software configuration module is in charge to install, configure, and start the application. These actions are done as modeled as in the Application Common Model.

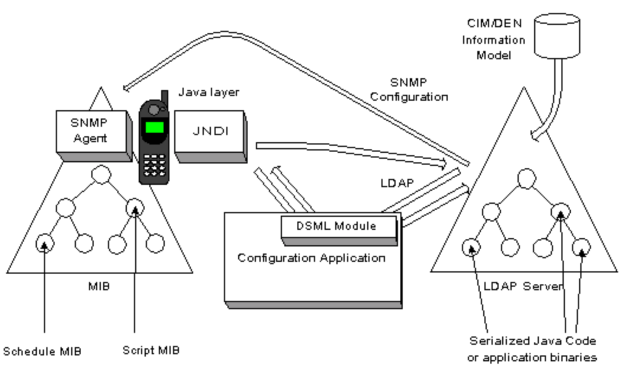

Fig. 3. Management architecture.

5. Conclusion

We have proposed in this paper a management architecture for the configuration of flexible and software-defined terminals. Our approach consists is extending the CIM/DEN to model radio software and to provide a Java-based architecture for the management of applications and their life-cycle.

Our work addresses the issue of providing an approach for managing wireless networks which is compatible with standard management of IP based networks. This is particularly interesting for the management of B3G networks, where end-to-end IP will be provided on top of wireless/core networks.

Service Management could benefit from our approach. We have now the capability to manage high-level services (video delivery offered by a service provider for instance) from this level up to individual radio resource needed.

Future work might consist in addressing fault management, since observing the failure of an application can be discovered with a good information model.

References

[1] DMTF (www.dmtf.org) Common Information Model (CIM) Core Model

[2] DMTF (www.dmtf.org) Common Information Model (CIM) Application Model

[3] J.Strassner. Directory Enabled Networks. Macmillan Technical Publishing. 1999.

[4] W. Bumpus, A. Westerinen et al. Common Information Model: Implementing the Object Model for Enterprise Management. Wiley. 1999.

[5] D. Verma. Policy-Based Networking: Architecture and Algorithms. New Riders Publishing. 2000.

[6] J. Mitola. Software Radio Architecture. Wiley. 2000.

[7] Joint Tactical Radio System. Software Communication Architecture Specification. 2001.

[8] D. Lund, B. Honary, K. Madani. Characterizing Software Control of the physical Reconfiguration Radio Subsystem. IST Mobile Submit. Barcelona, 2001.

[9] Moessner K, Gultchev S, Tafazolli R., Software Defined Radio Reconfiguration Management. Proceedings of the 12th International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC'01), San Diego, CA, USA, 30 September-03 October 2001.

[10] V. G. Bose, R. Hu, R. Morris. Dynamic Physical Layers for Wireless Networks Using Software Radio International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT May 2001.

[11] V. G. Bose and A. B. Shah. Software Radios for Wireless Networking. Infocomm *'98*, San Francisco, April 1997.

[12] P. Oommen. A framework for integrated management of mobile-stations over the air. Proc of 7th. IEEE/IFIP Symposium on Integrated Management IM2001. 2001.

[13] T. Howes, M. Smith, G. Good, T. Howes. Understanding and Deploying LDAP Directory Services. New Riders Publishing.1999.

[14] DSML. www.dsml.org

[15] Elliotte Rusty Harold, W. Scott Means. XML in a Nutshell: A Desktop Quick Reference (Nutshell Handbook). O'Reilly & Associates. 2001.

[16] Definitions of Managed Objects for Scheduling Management Operations. RFC 2591. IETF.1999

[17] Definitions of Managed Objects for the Delegation of Management Scripts. RFC.2592. IETF. 1999.

[18] Java Naming and Directory Interface. http://java.sun.com/products/jndi/