Software Reconfigurable Radio

Mei Yen Cheong

***Abstract*— The B3G mobile communications system is envisaged to provide radio environment with multiple co-existing radio access technologies and wireless standards. In order to ease the interoperability between wireless standards, provision of new services and smooth transition from one generation of mobile networks to another, the concept of reconfigurability is conceived. The reconfigurability concept provides mechanisms for both network and terminal equipments to dynamically select and adapt to available multitude of access technologies and standards. Various technical aspects that will enable software reconfigurable radio are discussed in this paper.**

***Index Terms*—End-to-end reconfigurability, Reconfigurability, Software defined radio, software download**

I. Introduction

Conventional hardware based radio systems have limited utilities since the parameters defining the modulation methods, waveforms, signal generation and link layer protocols are fixed. Due to the rapid evolution of wireless communications systems, from 2G to 2.5G to 3G, equipment manufacturers are facing the problems of equipment incompatibility because of different link layer protocols for different generation of networks. Furthermore, different wireless communications technologies are deployed in different countries or continents. These problems cause difficulties in the deployment of global roaming services and in rolling-out new services due to the wide-spread presence of legacy subscriber handsets.

Software defined radio (SDR) technology aims to overcome these problems by building an open-architecture based radio system software [1]. Functional modules of the radio system such as modulation/demodulation, signal generation, coding and link-layer protocols are implemented on generic reprogrammable hardware platforms using software.

The future generation or the beyond 3G (B3G) mobile communications systems are targeted to support integration and co-existence of multiple radio access technologies (RATs) in a common composite radio environment. An example of a composite radio environment is a single system that can be accessed by multiple access technologies such as GSM, GPRS, UMTS, WLAN, WIMAX and DVB. The concept of reconfigurability [2], which is an evolution of SDR, eases the implementation of the above environment. With the concept of reconfigurability, the mobile terminals and network elements can dynamically select and adapt to an appropriate RAT in a given service area at a given time. The reconfigurability feature for RAT selection is not only restricted to components pre-installed in the equipments. It also includes downloading, installation and validation of the software components needed for the reconfiguration. This way, equipment manufacturers can use a common design for multi-functional radios leading to increased market size for a single product. For network operators, the interoperability of different networks is enhanced and system upgrades and bugs fixing are easier to manage and implement. Finally, end users benefit from the enhanced functionality of their SDR device and the ability to achieve ubiquitous connectivity.

The SDR forum is a standards organization involved in developing specifications and standards ranging from definitions, architecture, software, hardware, software download protocols, security issues and many other aspects of SDR. The E^2R is a research project of the European Information Society Technologies (IST) [3]. The E^2R organization is involved in defining standards and specifications particularly regarding the end-to-end reconfigurability of SDR. This paper aims to discuss the aspects for successful implementation of software reconfigurable radio. Summary of related research results and findings of the SDR and E^2R are given in this paper.

This paper is organized as follows. Section II gives brief description of the SDR architecture. An overview of software download for RF reconfiguration is given in Section III. Section IV discusses the aspects of management and control architecture for the end-to-end reconfigurable equipment followed by conclusions in Section V.

II. Software Defined Radio Architecture

A. Hardware components architecture

Figure 1 shows the block diagram of a generic digital radio transceiver consisting of the radio frequency (RF) front-end, the intermediate frequency (IF) section and the baseband section.

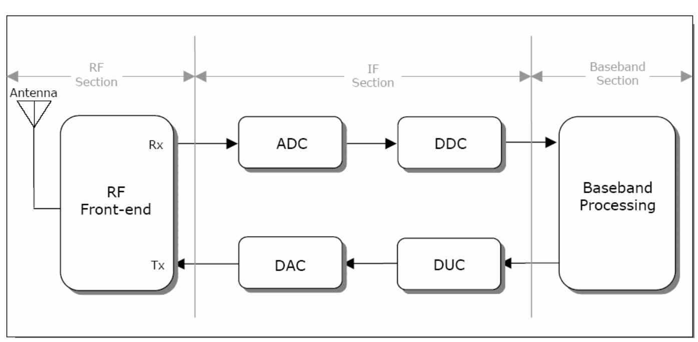

Figure 1. Architecture of a generic digital radio transceiver [1].

M. Y. Cheong Author is with the Signal Processing Laboratory of Helsinki University of Technology, Espoo, Finland (phone: 09-451 2935; fax: 09-452 3614; e-mail: meiyen.cheong@tkk.fi).

The RF front-end functions as the transmitter and receiver for the RF signal received via the antenna. On the receive path, it down-converts the RF signal to IF signal for further processing in the IF section. On the transmit path, it performs up-conversion to convert the IF signal to RF signal follow by power amplification.

The IF section is responsible for analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) on the receive path and the transmit path, respectively. The digital down converter (DDC) and digital up-converter (DUC) that proceeds and precedes the ADC and DAC respectively, jointly assume the functions of a modem.

The baseband section performs baseband operations such as connection setup, equalization, frequency hopping, timing recovery and correlation. In an SDR system, the baseband processing is designed to be software programmable. On top of that, the DDC and DUC modules in the IF section are also programmable. The link layer protocols, modulation and demodulation operations are implemented in software. Thus, the operational mode of an SDR enabled device can be changed or augmented post-manufacturing using software [5].

An ideal software radio system is one that is programmable up till the RF section, i.e., capable of performing high speed and power efficient analog-to-digital conversion and vice-versa right at the antenna. However, the current ADC/DAC technologies are not sufficient in supporting the required digital bandwidth, dynamic range and sampling rate for efficient implementation programmable RF section.

B. *Software components architecture*

The software components architecture of a typical SDR system is illustrated in Figure 2. The hardware resources layer is built on a generic hardware platform using programmable modules such as digital signal processors (DSPs), field programmable gate arrays (FPGAs) and microcontrollers (MCs), and analog RF modules.

The operating environment layer performs hardware resource management, memory management and interruption service. It also provides consistent interfaces to hardware modules used by the application layer above it.

The radio applications layer implements the link layer protocols and modulation/demodulation operations using software modules. The radio applications provide link-layer services to the higher level protocols such as WAP and TCP/IP.

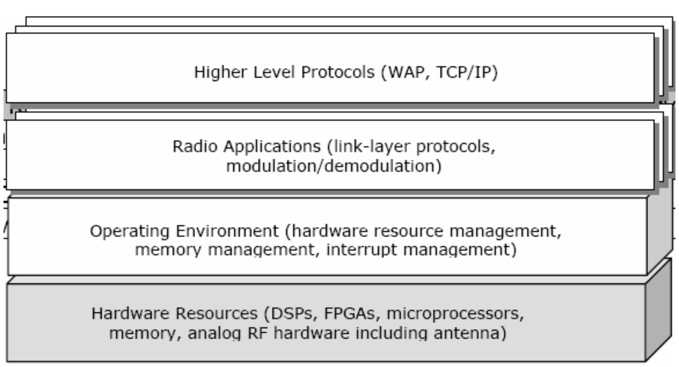

Figure 2. Software architecture of SDR [1].

III. Overview of Software Download for RF Reconfiguration

A. *Definition of radio software download*

The Software Defined Radio Forum (SDRF) [4] defines the term “radio software download” as the process of delivering reconfiguration data and/or new executable code to an SDR device to modify its operation or performance [5]. Software downloaded over the air such as free news, proprietary corporate data, email and multimedia materials are considered non-radio software. Non-radio software includes user applications and contents.

Radio software consists of primary radio software and ancillary radio software. Primary radio software directly affects the radio functionality. This software is tightly coupled with the radio hardware to derive the overall radio functionality. Ancillary software refers to radio software that affect the use of the device such as input/output driver and user interfaces. Both radio software and non-radio software can be further broken down into executable code and data. These characterizations are summarized in Figure 3.

Reconfiguration data consists of parameters for modulation techniques, power levels, operating frequencies and other operational parameters. Software program downloaded for enabling new digital signal processing algorithms, bug fixing and new radio air interface for SDR-enabled terminals and base stations are examples of executable code.

B. *Radio software download scenario*

The radio software download can be initiated by the network operator, the user or an application. There are two download environments, i.e., local download and remote download. Local download refers to data transfer through an attached cable, infrared, Bluetooth and technician at a kiosk.

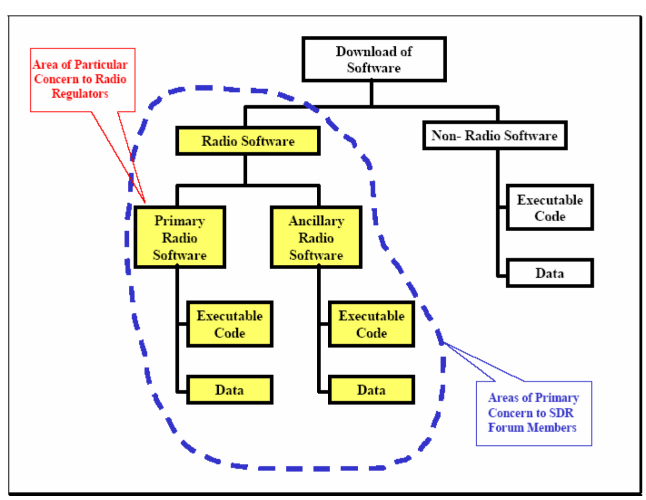

Figure 3. Characterizations of Software Download [5].

By contrast, remote download includes software download via the radio interface through the supporting core radio network. The downloaded software can be installed and activated immediately or saved for later reconfiguration. Software activation is a function of user and operator interactions associated with the download process.

Table I gives examples of eight download scenarios [5]. Each download scenario requires different security, authorization, authentication and other related considerations.

IV. END-TO-END RECONFIGURABILITY

The key objective of the E^2R project is to devise, develop, trial and showcase architectural design of reconfigurable devices and supporting system functions to offer an extensive set of operational choices to the users, application and service providers, operators, and regulators in the context of heterogeneous systems [3]. In order to fulfill these objectives, a series work packages (WPs), as shown in Figure 4, are proposed by the E^2R project to study all aspects, ranging from user device through system level, that make the end-to-end reconfigurable system possible.

In this section, the research results of the E^2R project on equipment management and control architecture for end-to-end reconfigurability [2] is summarized. E^2R results on end-to-end reconfigurability system requirements and architecture development can be found in [6] and [7] respectively. A series of white papers presenting research results of various WPs obtained in phase 1 of the E^2R project can be found in [8].

A. Capabilities of equipment

Equipments that are able to operate reliably and securely in an end-to-end reconfigurability context must possess the following capabilities.

- **Monitoring and discovering**
  The device must be able to periodically check for possible new RAT in the service area offering better opportunities such as higher QoS, lower cost per QoS level and service. This also includes collecting statistics from different RATs for status assessment.

- **Negotiating**
  The device must be able to negotiate offers with the available networks and selecting the most appropriate RAT.

- **Providing support**
  The device must be able to support various protocols and protocol features. According to the resources and capabilities of the device, dynamic insertion, replacement and configuration of protocols components from different vendors have to be allowed by the management and control architecture.

- **Verifying**
  The device must be able to verify the secure level of a download source prior to downloading and installation of software downloads. Appropriate checks such as authenticity, authorization and integrity check must be performed.

- **Controlling and coordinating**
  The device must be able to control and coordinate the reconfiguration of various equipment components.

- **Interacting**
  The device must be able to interact with external entities such as network entities.

B. Equipment management and control architecture

The requirements for equipment presented above trigger the needs for a generic management and control architecture. The E^2R proposed high-level structure of equipment management and control architecture is illustrated in Figure 5. The structure comprises of three main components, namely the configuration management module (CMM), the configuration control module (CCM) and the execution environment (ExEnv).

The CMM is a functional entity within the reconfigurable equipment. It is responsible for the management of all configuration tasks and negotiation of reconfiguration decisions with other entities. It manages the distributed controllers that initiate, coordinate and perform reconfiguration functions such as monitoring and discovery, software download, mode selection and switching, and security.

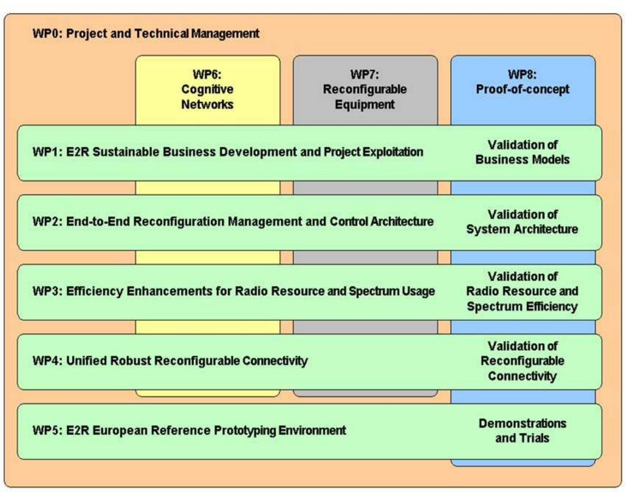

Figure 4. E^2R project structure

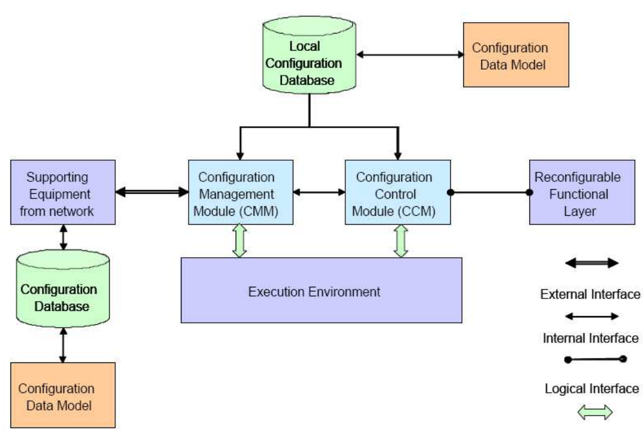

Figure 5. High-level structure of equipment management and control architecture [2].

TABLE I
EXAMPLES OF DOWNLOAD SCENARIOS [5]

Scenario	Description	Example
1	Remote download of air interface module and immediate mode switch, initiated by an application.	Use of allocation code to make decisions of required air interface software, based on user profile inputs. The application code requests appropriate download and completes download automatically and completes mode switch. This may involve negotiation between application software that has a user's profile and the network. This may or may not require immediate mode-shift and hard hand-off to another network.
2	Remote download of air interface module for later reconfiguration, initiated by application	Use of profile application code to automatically translate user's profile to download air interface module for subsequent use. May involve negotiation between application software and user's profile and network.
3	Remote download of air interface module and immediate mode switch, initiated by network	Switch air interface standard to adjust data pipe to appropriate bandwidth, possibly based on limited bandwidth availability or QoS considerations. Hard hand-off between modes may be required while transmitting data.
4	Remote download of air interface module for later reconfiguration, initiated by network	Download SDR software for version upgrade. Do not require immediate swiching. Could be general upgrade, or specific pathces to fix problem in handset.
5	Remote download of air interface module and immediate mode switch, initiated by user	Switch air interface standard to request specific service (higher bandwidth) or for new standard based on travels between areas with different air interface standards and service providers. This may require interaction between the user and network as user directly negotiates availability of certain standards with the network. This does not require interaction with applications that carries user's profile – user's request is adequately specific to bypass a profiler application. This may or may not require hard hand-off between modes while carrying traffic.
6	Remote download of air interface module for later reconfiguration, initiated by user.	Download air interface standard in one country before travel to another, but do not make immediate shift of air interface standard (that step is initiated separately). This does not require interaction with application that carries user's profile – user's user's request is adequately specific to bypass a profiler application.
7	Local download of air interface module for later reconfiguration, initiated by application	Application requests loading of specific air interface module. Cannot initially join cellular session because terminal is not configured. Application must request download of air interface module via connection to computer, via card or wireless standard like Bluetooth.
8	Local download of air interface module for later reconfiguration, initiated by user	User requests for loading of specific air interface module. Cannot initially join cellular session because terminal is not configured to do so. User must download air interface module via connection from computer, via card or wireless standard like Bluetooth.

The CCM is a supporting entity. It is responsible for the control and supervision of reconfiguration execution. Specific commands/triggers and functions of a given layer or execution environment are required to perform this. The functional modules of the CMM and CCM and their interactions are shown in Figure 6.

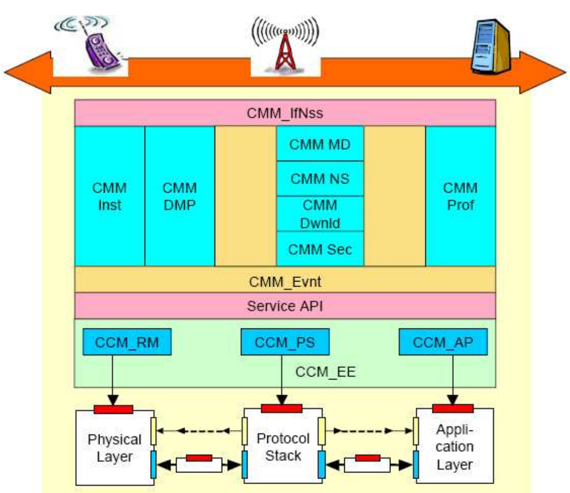

Figure 6. Structure of CMM and CCM.

The ExEnv supplies the CMM and CCM with a consistent interface for application of reconfiguration actions to the equipment. The ExEnv makes up the basic mechanism for dynamic, reliable and secure change of equipment operation. It consists of a set of interconnected hardware components and a set of software abstractions. The hardware can be both general purpose and application specific. Well-defined mechanisms and interfaces (Hardware abstraction layer (HAL) and proxy software components) are used by the software abstractions to access the hardware resources. This is depicted in Figure 7.

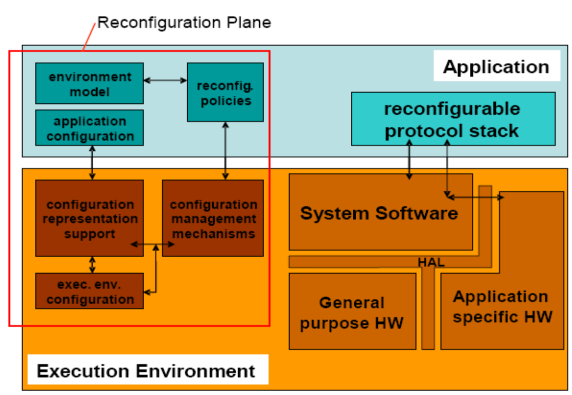

Figure 7. Reconfigurable execution environment

V. Conclusion

Software reconfigurable radio is a developing technology that involves studies of many aspects extending from user equipment to system level considerations. In order to meet the objective of using the reconfigurability concept in supporting the B3G rollout and operations, organizations such as the SDR forum and E^{2}R are formed to study and proposed the core strategy for its implementation. In this paper, the concept of reconfigurability using software download has been outlined. Several examples of software download scenarios are given. The equipment management and control architecture that are required for end-to-end reconfigurability are also discussed.

References

[1] Wipro technologies, "Software-defined radio, A technology overview," white paper, August 2002.

[2] J. Vogler, G. Pfeiffer, V. Stavroulaki, P. Demestichas, A. Katidiotis, D. Petromanolakis, N. Alonistioti, E. Patouni et. al., "Equipment management and control architecture," *End-to-End Reconfigurability* (*E^{2}R*) *white paper*, July 2005.

[3] IST-2003-507995 E2R Project, http://www.er2.motlabs.com

[4] Software Defined Radio Forum website, www.sdrforum.org.

[5] SDR Forum, "Overview and definition of software download for RF reconfiguration," SDRF-02-A-0002-V0.00, August 2002.

[6] E^{2}R, "End-to-end reconfigurability system requirements," *End-to-End Reconfigurability* (*E^{2}R*) *white paper*, January 2006.

[7] E^{2}R, "End-to-end reconfigurability system architecture," *End-to-End Reconfigurability* (*E^{2}R*) *white paper*, January 2006.

[8] E^{2}R white papers repository, http://e2r.motlabs.com/whitepapers/