Management and Controlling Architecture in E2E Reconfigurable Terminals

J. Brakensiek, D. Lenz, Nokia Research Center, Bochum, Germany
T. Wiebke, Panasonic European Laboratories GmbH, Germany
S. Gultchev, R. Tafazolli, University of Surrey, UK
A. Bisiaux, C. Moy, A. Kountouris, Mitsubishi Electric ITE-TCL, Rennes, France
M. Halimic, Panasonic Mobile Communications Development Lab, UK
C. Dolwin, Toshiba Research Europe Ltd, UK

1. Introduction

Reconfigurable terminals need a well-defined framework for the local equipment reconfiguration management to operate efficiently and reliably in an "end-to-end reconfigurability" context.

A distinction is made between two complementary aspects of the configuration management issue: functionality and architecture. Both depend on the requirements we place upon the reconfiguration process, which are dictated by the usage scenarios.

A reconfiguration management architecture and functionality will be introduced that will offer interfaces to external network entities and control the internal components to select, negotiate and perform the appropriate reconfiguration actions.

This will require a distributed reconfiguration manager architecture with one equipment management entity for the communication to the network system entities and a reconfiguration controller for control and co-ordination of the equipment components on the application layer, the flexible protocol stacks and reconfigurable modem.

2. Re-configurable Terminal

Specifically the Modem Reconfiguration process is addressed involving the Equipment Reconfiguration Manager and the Modem Reconfiguration Controller as shown in Figure 1.

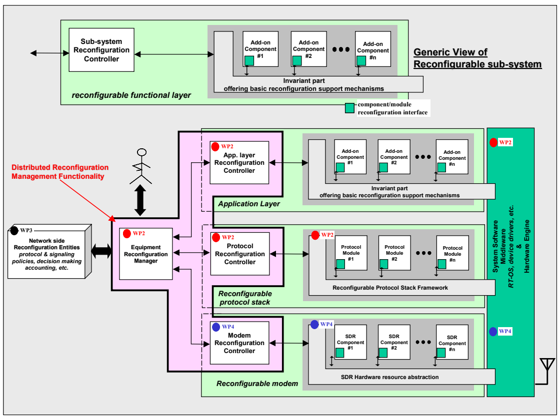

Figure 1: Overall reconfigurable equipment architecture

Reconfiguration inside a terminal in an end-to-end manner, as considered in the E^2R[5] project, is affecting all layers from the physical up to the protocol and application layer. In this paper we are focussing on local reconfiguration interaction between the reconfiguration management modules and the control elements within the physical layer.

2.1 Local Reconfiguration Management

The local reconfiguration manager will provide procedures/functionalities grouped into the following categories to impose, select or assist in the selection of reconfiguration actions:

(1) Control of equipment components to coordinate the realisation of a reconfiguration that requires modifications of various equipment components,
(2) General supporting procedures, namely, software download and installation,
(3) Monitoring and discovery of the capabilities, status and offers of the networks in a certain area,
(4) Evaluation of hardware/software processing capabilities and constraints of the reconfigurable components,
(5) Reconfiguration action selection and negotiation to select appropriate reconfigurations based on QoS requirements of applications and users, and the network capabilities.

All the functionality needs to be considered for the local interaction between the management module and the component control module such as the reconfigurable physical layer. The Configuration Management Module (CMM) is responsible for this local interaction to ensure the end-to-end reconfigurability context.

2.2 Reconfiguration Control in Physical Layer

Reconfiguration of the terminal's system functionality will be based on a set of re-configurable, programmable or parameterizable hardware/software resources available for doing physical layer data and control processing. Having this in mind, a dedicated module is required, which takes over responsibility for the configuration process. In particular the following functionality is required:

- Control and supervision of the re-configuration process

- Definition of fallback modes, in case of mal function
- Provision of download capability for complete or partial configuration data update

This functionality will be located inside the Configuration Control Module (CCM) of an end-to-end reconfigurable device.[4]

3. Distributed Configuration Management

The functional aspects were previously described in terms of the local reconfiguration related procedures that need to be considered. In this section the adopted equipment configuration management architecture within end-to-end reconfigurable networks is presented. The focus is on terminal design aspects to enable the terminal to provide for reconfiguration (run-time or not). Even though the ensuing discussion is mainly user-equipment (i.e. mobile terminal) oriented, with proper adaptation the presented concepts are applicable to basestation and access point equipment as well.

Architecture relates to the distribution of the different functions across the network and their organization as logical entities collaborating to achieve reconfiguration[6].

3.1 Requirements and logical partitioning

A distributed hierarchical configuration management approach is adopted. It is distributed in the sense that in an end-to-end reconfigurability context many physically or logically distributed entities collaborate. It is hierarchical since a configuration can have an abstract representation to the entities outside the equipment, e.g. GSM or UMTS configuration. Inside the equipment this representation becomes more and more detailed as we approach the real hardware that needs to be configured.

Our basic assumption is that inside, the terminal equipment is organized as a stack of reconfigurable layers.

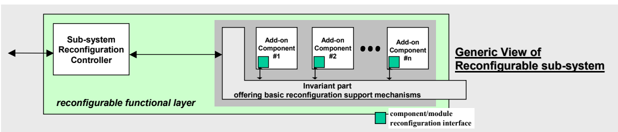

Figure 2: Generic View of Reconfigurable Subsystem

As shown in Figure 2, a reconfigurable layer consists of a component framework on which the various components are composed together. This framework can be considered as part of the equipment execution environment. In terms of reconfigurability this infrastructure provides basic support for component plug and play. A Layer Reconfiguration Controller is responsible for the instantiation of a new configuration within the layer. This entity has access to a detailed representation of the current layer configuration, it has access to configuration data related to potential configurations and finally uses the configuration related APIs provided by the component framework to instantiate a new configuration. The interaction between the controller and the framework is defined by means of an internal reconfiguration control interface.

3.2 Equipment Reconfiguration Manager

Referring to Figure 3 we can see that once these reconfigurable layers are stacked together their Layer Reconfiguration Controllers can be viewed as a vertical reconfiguration layer. Their collaborative action is orchestrated by a supervising entity called the Equipment Reconfiguration Manager. Interactions of the equipment reconfiguration manager with the layer reconfiguration controllers are defined by means of internal configuration management interfaces. This entity is also responsible for the interaction with reconfiguration related entities situated within the core-network. This interaction is defined by means of an external network reconfiguration management interface.

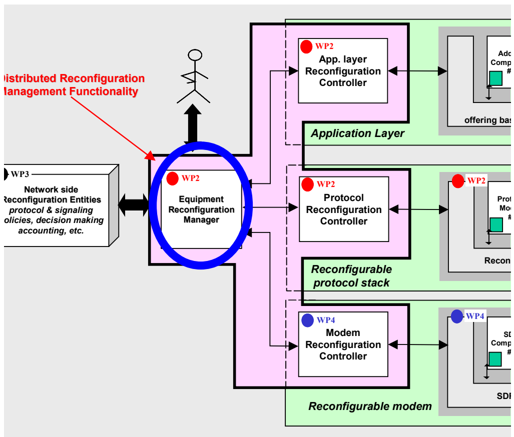

Figure 3: Distributed Reconfiguration Manager

For the reconfiguration process the network may be more or less involved depending on the usage scenario (see discussion on scenarios later on) and the equipment capabilities for autonomous decision-making. Network related aspects (e.g. network reconfiguration manager, reconfiguration protocol) are out the scope of the present paper and will only be superficially highlighted in order to define the equipment operational context.

In order for the equipment reconfiguration manager (EqRM) to perform reconfiguration involving multiple layers or even some layer specific it needs to establish interaction with multiple layers. The next section provides some explanation concerning these interactions.

4. Configuration Control

The configuration control module (CCM), which is identical with the modem reconfiguration controller from the EqRM point of view, is responsible for the physical layer related configuration.

4.1 Requirements and Logical Partitioning

From Figure 4 we can see that the CCM has two main interfaces. The first interface is with the CMM. This interface is used by the CMM to negotiate a configuration with the CCM. In return the CCM will request Data Processing Modules for reconfigurable elements such as a DSP or reconfigurable logic.

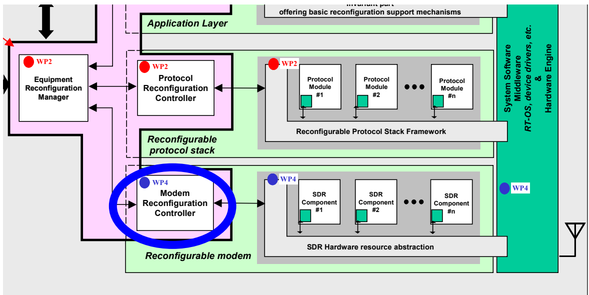

Figure 4: Configuration Control Module

The second interface is between the CCM and the hardware resources and supports the downloading of object code to reconfigurable resources such as DSP's and reconfigurable logic as well as the setting of parameters for hardware accelerators. In addition to configuring the data processing elements the CCM will also manage the communication resources to ensure links between the processing elements maintain the required throughput.

Besides that the assigning of functions to resources is referred to as spatial scheduling but in addition there is also a requirement for a temporal scheduler that is responsible for defining when a resource will execute a function. The temporal scheduling is implemented using the services of the *Operational Software* module. The *Operational Software* is responsible for orchestrating the resources to ensure the real-time constraints are met. The CCM exists in the logical domain i.e. uses services supplied by RTOS, device driver, etc. rather than driving hardware directly.

Therefore the key requirements for the CCM can be defined as:

- Independence from the hardware resources
- Dynamic download of Data Processing Modules
- Power conscious
- Standard interface to the CMM
- Seamless and partial re-configuration
- Reliable, predictable and secure configuration
- Configuration and control of the receive and transmit path and associated signal processing applications and security functions

4.2 CCM Functionality

As noted above the CCM has to take the responsibility for reconfiguring the underlying physical layer. To afford this the CCM should implement various tasks, which can be logically partitioned into the following modules, as depicted in Figure 5.

- Authentication & integrity check of configuration
- Negotiation with CMM
- Memory management
- HW resource management & spatial scheduling
- Monitoring of environment
  - Statistics (used resources, status information)
  - Metrics (SNR, multi path, interference)
- Autonomous low-level adaptation to changing physical conditions

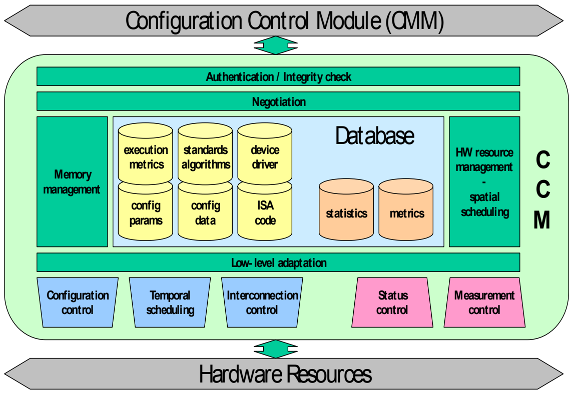

Figure 5: CCM Functional View

- Maintenance of the databases containing
  - Execution metrics
  - Standards and algorithms
  - Configuration parameters
  - Configuration data
  - SW modules (ISA code)
  - Device driver
  - Gathered statistics & metrics

By integrating these processes the CCM should be able to fulfil the requirements and to ensure proper operation of the underlying hardware resources.

4.3 HW resource configuration

Reconfigurable architecture of the physical layer consists of a set of reconfigurable hardware (analog and digital) resources. In the process of reconfiguration, these hardware components are interconnected, monitored and controlled by the CCM. For each hardware component the CCM has to have information about data, control and reconfiguration interfaces in order to exchange information such as

- Reconfiguration time
- Amount of reconfiguration data
- Partial vs. full reconfiguration
- Parameter range
- Reference for the execution time
- Reference for the power consumption

Since the type and level of reconfigurability of hardware resources vary from the truly general-purpose components such as Digital Signal Processors or Field Programmable Gate Arrays to dedicated components such as programmable oscillators there is a need for a generic view of hardware resources. Therefore, hardware resources are modeled in a uniform abstract way providing required information to the CCM and hiding the implementation specific details. In addition, this approach enables a seamless integration of a new type of hardware into the architecture.

5. CMM/CCM INTERFACE

5.1 Basic Interaction between EqRM/CMM and CCM

Let's examine into some more detail the interactions of the EqRM with a specific reconfiguration-layer like for instance the PHY layer. The EqRM talks to the Modem reconfiguration controller via the CMM interface. The CMM entity is the part of the EqRM responsible for the interactions with the physical layer. These interactions mainly consist, from the EqRM/CMM point of view, of:

- Collection of environment PHY related monitoring data
- Collection of equipment operation monitoring data
- Transfer of PHY configuration data
- Reconfiguration command issue
- Acknowledgements

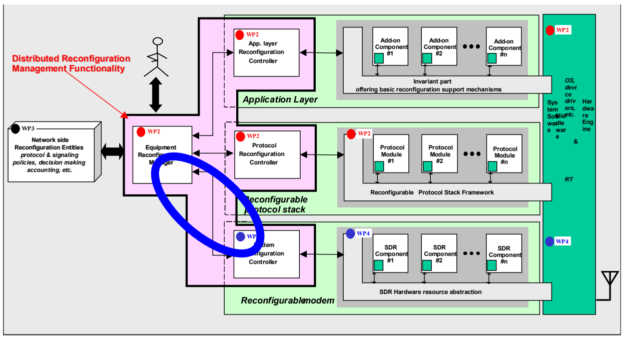

Figure 6: EqRM and CMM interaction

At each layer statistics that permit the identification of the environment must be collected. For the physical layer such information can relate to the channel conditions and the radio link quality, the velocity or even position. Also, appropriate statistics that permit the identification of equipment operation problems can be gathered as well. Some of this information needs to be passed to the CMM either for the reconfiguration action selection or for further transmission to some network entity. Similarly if the EqRM receives configuration data the part concerning the physical layer has to be transmitted by the CMM to the CCM. Finally, the EqRM will be responsible for sequencing and triggering the required reconfiguration commands; such commands may result either from a network or locally initiated requests. Those commands involving the physical layer will be issued by the CMM.

5.2 Control and Management Framework

The framework for control and management interfaces will depict the control interfaces and procedures of the CCM to link to the reconfiguration management part. This will involve the specification of the control and data messages that are to be exchanged between them. In particular, it will lead to the development of the protocols for communication between the control and management parts that will carry these messages.

This framework will present a generic approach for reconfiguration procedures to fit not just a full reconfiguration of the modem but also of single modules (part of the modem) or even sub-modules. In practical terms, the CMM/CCM interaction can be regarded as a system based on a dual state machine mechanism with translator in between allowing the correct interpretation of each state machine's state.

In this respect, the structural entity of the terminal reconfiguration management and control will stay untouchable and interchangeable. This will allow the flexibility of using different hardware and software platforms and systems and not to have the need to modify the reconfiguration platform.

6. Scenarios

The role of the entities involved in the reconfiguration process as well as their interactions are illustrated here through one example. It relies on a basic reconfiguration scenario relating to intra-standard performance enhancement, which is considered as one of the first achievable case-studies in E^2R's roadmap.

This scenario involves the UE and, to a certain extent, part of the network. It consists in downloading a new piece of code (or patch) to either replace a bug in an algorithm of the modem or improve its performance[7]. The reconfiguration process can be divided into the following consecutive steps:

- First, a dialog must take place between the UE and the network reconfiguration manager (NRM) so that the UE can be informed that a new patch is available for download. This implies that the UE fully identifies itself (reference of its hardware and software, versions of the algorithms that may be upgraded).
- The NRM then selects the appropriate patch and sends it to the UE as a classic payload through a normal communication standard that supports reliable and secure data transfers (this standard may or may not be the one that is being updated).

- The UE recognises the payload as configuration data and installs it in its local configuration database; it is now ready to use the updated version of the code.
- Even though the patch is downloaded while on-line, no specific real-time constraints need be met and actual reconfiguring may happen off-line. The update will be effective next time the corresponding standard is instantiated (at boot-up or when switching from another standard) so that no conflict arises if the standard that is used to download the patch is the one that is being updated.

The operations required to perform such a reconfiguration scenario are distributed among the NRM, CMM and CCM in the following manner:

Role of the NRM

- The NRM has access to a patch database provided by the UE manufacturer.
- It maintains a dialog on a regular basis with the UE in order to inform it when a new patch is available.
- It also has the capabilities of a classic secure server that the UE can connect to for downloading the patch.

Role of the CMM

- The CMM manages the dialog with the NRM by sending to information on the hardware and software parts that are used in the physical layer.
- When a new patch is available, it takes the decision to either download it immediately, postpone it until a more suitable time (e.g. because the UE is currently too busy) or even decline (e.g. because the UE's configuration memory is full).
- The CMM is capable of opening a new communication channel to download the patch without interfering with already active channels.
- The CMM passes the patch on to the CCM.

Role of the CCM

- The CCM keeps track of the various algorithm versions that it possesses in configuration memory.
- It provides the CMM with all the information needed on those algorithms as well as the underlying hardware and software components that run them.
- Once the patch has been downloaded, the CCM installs it into its configuration memory; the previous version is kept for rollback capability.

7. Conclusions

Terminal reconfiguration in an end-to-end manner requires a management and controlling architecture, taking care of the physical layer as well as the protocol layer and execution environment configuration. A basic functional split of a distributed management and the controlling entities has been derived. Definition of interfaces is essential in order to achieve a platform and implementation independent solution from the hardware resources perspective as well as from the operational software environment. An example scenario is highlighting the functional partitioning between the CCM and CMM. The CCM is the entity, which has access to all physical layer resource, providing metrics and statistics to the CMM. The CMM is responsible for providing the correct configuration instructions, based on the provided information.

Therefore efficient and reliable CCM/CMM functionality and interaction is the key for the successful implementation of a software reconfigurable terminal.

8. Acknowledgments

The work reported in this document forms part of the EU sponsored IST 6th Framework Project End-to-End Reconfigurability (E2R, IST-2003-507995), the contributions of both EU and project partners are gratefully acknowledged.

9. References

[1] J. Pereira, *Re-Defining Software (Defined) Radio: Re-Configurable Radio Systems and Networks*, IEICE Trans. on Comm., vol. E83-B, no. 6 pp. 1174, 2000.

[2] J. Mitola III, G.Q. Maguire Jr.., *Cognitive Radio: Making Software Radios more Personal*, IEEE Pers. Communications , vol. 6, no. 4, Aug. 1999, pp. 13-18.

[3] K. Moessner, S. Gultchev, R. Tafazolli, *Software Defined Radio Reconfiguration Management*, in Proc. 12^{th} PIMRC'01, vol. 1, pp. 91-95, USA, Sept. 2001.

[4] J. Brakensiek, M. Darianian, S. Walter, C. Dolwin, M. Halimic, L. Maurer, A. Kountaris: *Reconfigurable Physical Layer Architecture supporting End to End Reconfiguration*, WWRF Meeting New York, October 2003.

[5] End-to-End Reconfigurability (E^2R), IST-2003-507995 E^2R, http://www.e2r.motlabs.com

[6] A. Kountouris, C. Moy, *Reconfiguration in Software Radio Systems*, 2nd Karlsruhe Workshop on Software Radios, Mar. 2002.

[7] C. Moy, A. Kountouris, A. Bisiaux, *HW and SW Architectures for Over-The-Air Dynamic Reconfiguration by Software Download, SDR Workshop of the IEEE Radio and Wireless Conference,* Boston, USA, Aug. 2003