SUBMISSION TO THE
SDR FORUM 2007 TECHNICAL CONFERENCE

TITLE:

VERIFICATION PROCEDURE FOR RECONFIGURABLE TERMINAL CONFIGURATIONS

Authors:

Stoytcho Gultchev, Klaus Moessner (University of Surrey, United Kingdom)

Corresponding Author:
Stoytcho Gultchev (The University of Surrey, GU2 7XH Guilford, United Kingdom)
Tel: +44 1483 683 605, Email: s.gultchev@surrey.ac.uk

VERIFICATION PROCEDURE FOR RECONFIGURABLE TERMINAL CONFIGURATIONS

Stoytcho Gultchev, Klaus Moessner
Mobile Communications Research
Centre for Communication Systems Research,
University of Surrey Guildford, Surrey, UK - GU2 7XH
(S.Gultchev, K.Moessner@surrey.ac.uk)

ABSTRACT

SDR technology is becoming integral to link different wireless technologies, and to enable interoperability between them. This raises the question of the need for SDR related standards and suitable regulation allowing the use and deployment of SDR technology. Standard compliance and a system's correct functioning are generally the most critical problems and they need to be ensured. Addressing conformity and correct functionality, software and configuration verification and validation techniques are needed. These important requirements have to be agreed and satisfied so that SDR equipment facilitates interoperation between different radio access systems and the various types of reconfigurable platforms available. This paper gives an overview of the influencing factors and provides some direction and recommendations of how compliance can be achieved. This includes a description of the approach that is taken by the E^{2}R project and that can be applied to ensure compliance between different adaptive communication platforms.

1. INTRODUCTION

To enter the European market, the different manufactures have to follow the R&TTE Directive, which requires them to verify the compliance of equipment to a set of harmonized standards and to self-certify the compliance of their different HW/SW configurations. For the market introduction of SDR terminals, a sort of class marking approach [1] has been proposed to simplify the ways in which compliance can be ensured. However, it is conceivable that third party SW providers and uncountable combinations of software modules and configurations will come into existence. The classmark approach would not be sufficient to ensure standard compliance in this case; neither would the current R&TTE regime suffice to handle this (i.e. currently the R&TTE Directive does not foresee a certification of software only).

However, if SDR equipment should be operated in an open manner, allowing the use of configuration software obtained from different sources, thus shaping an open software environment, the need to facilitate standards' conformance becomes immense. To ensure that the system specifications of the target RAT are met, in such open SW regime, radio configurations (HW/SW combinations) have to be tested and validated before they can be brought to function. Therefore, robust mechanisms are needed that will allow validation not just when the terminal is connected to the home network, but also in situations where a reconfiguration takes place when a user is in a visiting situation.

A conformity verification scheme has to be sufficiently flexible to be used beyond administrative and regulatory boundaries. E^{2}R [2] defined a so called Virtual Procedure (VP), which can be used to verify the compliance to harmonised standards as well as the fulfilment of essential requirements and the actual certification of hardware/ software combinations of reconfigurable terminals.

Task of the VP is to detect possible violations of harmonised standards *before* the actual start of a reconfiguration sequence. It allows the evaluation of an intended configuration thus providing the possibility to abandon a reconfiguration process if required. The VP, its functions and the sequences forming the verification process have been developed and implemented. To be able to test the compliance of a RAT implementation in the VP, a unified system design approach has to be followed during the design phase of a RAT. The approach requires markers to be able to implement verification and validation of the single configuration modules.

This paper provides a description of the mechanisms, which facilitate the pre-installation testing of configurations. A two stage approach, first evaluation of interface compatibility and second functional testing, are described. The paper includes a description of example modules as well as the definition of the actual test-procedures executed in the VP.

The main principles and initial demonstrator set-up are described. The design is presented using SDL (Specification and Description Language) and the design verification using SDL tools is documented.

2. REASONS FOR VALIDATION OF RECONFIGURATIONS

The 'reconfigurability architectures' are defined to handle download, control and installation of the targeted radio configurations and have the main aim to support mechanisms, which ensure the correctness of the targeted configurations (i.e. a configuration being the HW-SW bundle implementing a radio). Any such software installation will be prone to the usual issues like implementation, viruses, security threats etc. Secondly, the verification of these configurations and their (potentially destructive) effects on the radio emissions and, if something goes wrong, the question of the responsibilities has been widely discussed. Thirdly, the network operators will be concerned about how to ensure that a SDR node is properly installed and won't negatively affect or bring down their infrastructure, or create interference to neighbouring or co-located systems.

Manufacturers will ensure the correctness of the initial SDR architectures and reconfigurable systems at the time of delivery. In the long run, due to the expected fragmentation of the market (new manufacturers, new software vendors) and the increase in availability of reconfigurable systems, there will be a need to deploy new procedures for verification of mixed configuration. This complexity of configurations cannot be just classified in a specification as documented in the E^{2}R class marking approach [1]. It will require much more to reflect the whole reconfiguration validation process, which may be based on the class mark mechanism, but even configurations outside such framework may be implemented in reconfigurable nodes; then testing the configurations, heir implementations in the network, with the necessary platform interaction for control and management will be part of the verification process. A variety of factors will influence this validation process but the most important will be portability, interoperability, spectrum efficiency and compatibility between platforms and reconfigurable systems. The result of a complex interaction between different players as part of it will only be achieved by standard compliance of a common reconfigurable framework. In this process, as described in [3], following issues need to be covered:

- Common interface definition;
- Standard compliance to the functionality of reconfiguration plane [4];
- New tools for system software reconfiguration modules' development;
- Common reconfiguration procedures of process handling download, control and manage reconfigurability;
- Deriving standard comprehensive verification mechanisms like:
  - ♦ A **standards (RAT) validation procedure** on the network side;
  - ♦ A **reconfiguration service provisioning mechanism** to be matched with specific (generic) network topology [5];
  - ♦ Ensuring **the reconfiguration node security** and security provisioning support [6]. Reconfiguration procedures will require verification and validation process techniques to ensure the intact of the reconfiguration security system of the reconfigurable nodes.

All these factors lead to a need for specification of the verification and the validation procedures in all the above cases.

To achieve a system where software reconfigurable nodes fully comply (to the applicable specifications) when reconfigured will need a systematic approach for every area mentioned above with consensus between reconfiguration service and equipment providers on a generic overall reconfigurable architecture.

This standard compliance can be captured and achieve in an unified system design approach which considers, already during the design phase, the issue of verification and validation of the single software modules, but also of the complete configurations. This uses a design suite that employs UML for the overall system and functionality definition, and SDL for the design of the information flow and verification mechanisms of the system.

3. VERIFICATION PROCEDURE

The principles and mechanisms outlined in [3] were followed when designing and evaluating the Verification Procedure (VP). The solution to the main concerns how, in an open software environment, radio configurations can be tested and validated without having to go through the tedious and time consuming type approval processes, the principles and mechanisms from [7] were adopted. A validation procedure has been included in the network counter part of E^{2}R is defined to produce information and knowledge about the reliability of the intended radio configurations. This includes an entity **VP manager** responsible for the performance of the Virtual Configuration Procedure. It contains a database to store temporally the software and virtual validation tool used for performing the testing of configurations on the bases of their profiles. It also performs initial testing of received software and profiles before they are sent to the terminal for execution. The network part also contains a Rules and Policies tool, which specifies platform dependent parameters and network dependent reconfiguration policies. Reconfiguration Software store is used as a database hosting approved configuration software and terminal configuration register.

The VP is designed to detect possible violations of radio standards during verification of the reconfiguration sequences and to validate whether the intended configuration complies with the given standards.

There are different stages where the standard conformance has to be tested during reconfiguration. Starting from the upload of new configuration software modules to the software repository, through to the installation and implementation of a piece of software the validity has to be evaluated. This also includes the download of the software modules and associated policies and requires mechanisms to confirm/ensure the integrity of the downloaded code.

Before this operation is executed the network performs a test of the downloaded software that confirms compliance to the initial specifications of the test case (i.e. that may be provided with the software) and compares the I/O parameters of the tested module with the margins provided (with the software). This validity information is than stored in the network server and the terminal can download/use the modules software for configurations. This mechanism provides the merging point for the reconfiguration software with the standards and provides the test results together with the software and policies.

The next stage is to download the software to the terminal and to ensure that the software is correctly stored in the E^2R terminal database. This download also applies when software or rules are required in the terminal for installing and creating the FDL description [8] of the terminal 'blueprint' respectively.

The third step is the testing of the FDL description, the file is downloaded to the VP, which in turn evaluates the tag-files compliance to the network requirements. This is one of the most important verification steps for the new terminal configuration; this has to be completed before a reconfiguration can take place within the terminal. Once the procedure is finished, the reconfiguration procedure takes place as the confirmation is send back to the terminal.

Finally, the installation of a module on the radio platform takes place and the terminal performs a final test before the different radio modules are connected and the new configuration becomes active. Verification procedures may differ, depending on the type/class of reconfiguration process; hence a number of reconfiguration scenarios are applied to demonstrate the functionality of the validation procedure.

The SDL implementation formally specifies the design of the modules in the Verification Procedure, and the communication sequences between the modules. With the SDL specification, the Verification Procedure is formally modeled by executable specification, which can be simulated and validated before the architecture is implemented and coded darkly. Through the SDL specification, the architecture is verified and validated in an early stage, which is valuable for complex system design and save the cost of software development.

In practice the testing and validation are closely related, and after the SDL system is debugged well enough, there is no further testing to be done. The scenarios are already tested and shown in the simulation MSCs. In the similar manner networked entities of the Verification Procedure are evaluated for the performance of procedures and message sequences compliance. This activity also deals with the correctness and validation of the design of the mechanisms between different network support entities.

In this work, state space exploration and state transition diagrams are used for validation of the design specification, and this is a well-known technique for automatic analysis. For the software terminal model is used a SDL Validator to do the validation task. All the modules' state transition diagram from the SDL Validator, applied to the Verification Procedure can be found in [3], the paper also shows the states of the modules, and what signals trigger the modules to action and the current state changes to another state are omitted from this paper.

The SDL specification for the soft terminal model is debugged and revised according to the results from the simulation and the validation. The specification finally becomes complete and correct, with no deadlock and no starvation, at least from the SDL specification point of view. The symbol and transition coverage of the specification are both 100%, which means the state space of the system is reachable and explorable completely.

4. SUMMARY

Network support services for terminal reconfigurability provide complementary mechanisms for provision of reconfiguration verification processes. In this paper the mechanisms described provide the possibility for network providers to control the reconfigurability of software definable equipment and to ensure that the possible configurations comply with given radio access standards and requirements. The paper shows the dependencies between the different parties involved and how they can provide reconfiguration validation, control and support to the reconfigurable terminal.

The paper presented the most important factors for reconfiguration verification and different verification techniques that need to be deployed for the uninterrupted and correct functioning of a reconfiguration node. The different verification methods have been highlighted to underline the importance of FDL description in validation of reconfiguration systems.

Finally, an example of the implementation and verification of reconfigurable radios and Verification Procedure was introduced as part of the E^2R terminal verification process.

5. ACKNOWLEDGMENTS

This work was performed in project E^{2}R II which has received research funding from the Community's Sixth Framework programme. This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein. The contributions of colleagues from E^{2}R II consortium are hereby acknowledged.

6. REFERENCES

[1] Berzosa F, "Reconfiguration Terminal Classmarks", WWRF 14, San Diego, California, USA, 07-08 July 2005

[2] IST project E²R http://e2r2.motlabs.com/

[3] Gultchev S, et.al. “Verification of software reconfiguration platforms” SDR05 Technical Conference, Orange County, California, USA, 14-18 November 14-18, 2005

[4] Gultchev S, et.al. “System Reconfigurability” SDR Technical Conference 2004, Phoenix, Arizona, USA, 15-17 November 2004

[5] Gultchev S, et.al., “Reconfiguration Mechanisms and Processes in RMA controlled Soft-Radios Signalling”, SDR Technical Conference 2003, Orlando, Florida, USA, 17-19 November 2003

[6] Gultchev S, et.al., “Provisioning of Reconfiguration Services between Different Access Networks”, published in Frequenz 58 (2004) 5-6, ISSN 0016-1136, pp.126-131, May-June 2004

[7] Gultchev S, et.al., “Network Based Reconfiguration Support Services for Software Radio Terminals”, IEE 3G2003, London, UK, 25-27 June 2003

[8] S. Zhong, C. Dolwin, R. Burgess, “A Sofware Defined Radio Proof-of-concept Demonstration Platform”, SDR Forum Technical Conference, Nov 2006