A SIMULATION-BASED APPROACH FOR PERFORMANCE EVALUATION OF SDR BASEBAND ARCHITECTURES

Anthony Barreteau, Sébastien Le Nours, Olivier Pasquier (Lunam Université - Université
de Nantes, UMR CNRS 6164 Institut d'Electronique et de Télécommunications de
Rennes (IETR), Polytech'Nantes, Nantes, France; anthony.barreteau@univ-nantes.fr ,
sebastien.le-nours@univ-nantes.fr, olivier.pasquier@univ-nantes.fr).

ABSTRACT

Application of SDR concept for mobile device presents numerous opposing requirements including low power, low cost, high performance that create significant design challenge. In this context, system architect is supposed to design SDR baseband architecture able to support a wide variety of communication interfaces with a wide range of processing resources that fully meet these requirements. Thus, it becomes necessary to help system architect to analyze and compare the growing number of potential architectures. The creation of efficient executable models becomes a mandatory step to enable architecting of such systems. In this paper, we present a simulation-based approach for performance evaluation of SDR baseband architectures. This approach makes possible to evaluate by simulation the expected resources according to complex use-case scenarios. This approach is illustrated through the study of an adaptive multi-standard and multi-application radio communication system.

1. INTRODUCTION

Current trends in the design of radio communication systems consist in offering terminals able to easily roam among heterogeneous networks like cellular, wireless local and metropolitan area networks [1]. The purpose of these improved functionalities is to increase data rate and to propose access to a wide variety of services anytime and anywhere with a single device. As a result, the number of protocols to be supported by a single receiver, and the number of modes supported by a single protocol, calls for higher flexibility. This implies to design multi-standard and multi-application radio communication systems able to adapt to their changing environments and to make decisions about their operating behavior to fully meet expected quality of services (QoS). The inferred complexity of hardware and software resources of terminal architectures is then significantly increased in order to support such advanced services.

Currently, parallel architectures clustered by application category are adopted to implement mobile terminals [2]. These multi-core platforms consist of a set of modules like fully programmable processor cores, standard interface modules, memories and dedicated hardware blocks. The definition of such platforms is done in order to fully meet functional requirements, related to the protocols to implement, and non-functional requirements such as timing constraints, power consumption, and cost. In the process of system architecting, potential architectures are compared according to performances achieved. Performance evaluation of candidate architectures early in the development process has then become mandatory in order to avoid costly design iterations. Typically, the creation of architecture models for performance evaluation is done by mapping a model of the system application onto a model of the considered platform [3]. The resulting description is then evaluated through simulation or analytical methods. Compared to analytical methods, simulation approaches are required to investigate dynamic and non deterministic effects in the architecture model. Architecture models are then simulated to evaluate usage of resources with respect to a given set of stimuli. Simulation results are used in order to compare performances of candidate architectures. In the context of next generation communication systems, one of the challenges is to design architecture able to implement baseband processing related to various communication interfaces supported that meet power consumption requirement and costs constraints. Current trend consist in defining SDR baseband architecture made up of wide range of processing resources to meet functional and non functional requirements of these systems. SDR baseband architecture is closely linked to changes of the system and its radio environment. It then becomes mandatory to provide means to capture at a high abstraction level to improve evaluation of various operating scenarios.

This paper presents an approach for performance evaluation of SDR baseband architectures. A modeling technique based on scenario files is presented to drive the behavior of the modeled environment in order to facilitate evaluation of different use cases. Architecture and its environment are described graphically through a specific activity diagram notation. Modeling techniques are presented to capture new requirements identified in adaptive multi-standard and multi-application radio communication systems. We also present techniques to efficiently model the various protocols supported and their related management. The behavior related to communication interfaces is described as a finite-state machine to model the use of platform resources when application executes. This model is then automatically generated in SystemC to allow simulation and to compare performances of various architectures. Simulation results are provided to analyze time properties and architecture performances according to different use-case scenarios.

The remainder of this paper is structured as follows. Section 2 analyzes the existing modeling and simulation approaches for performance evaluation of system architectures with a specific focus on the study of radio communication systems. In Section 3, the proposed modeling approach is presented and related notations are defined. In Section 4, we describe the considered case study and we detail the modeling techniques proposed to capture the different services supported. In Section 5, we present application of the proposed modeling approach to evaluate performance of a SDR baseband architecture. Finally conclusions are drawn in Section 6.

2. RELATED WORK

Performance evaluation of embedded systems has been approached in many ways at different levels of abstraction. A good survey of various methods, tools, and environments for early design space exploration is presented in [4]. Related approaches mainly differ according to the way application and platform models are created and combined. In the context of this paper, we especially focus on related works about performance evaluation for flexible terminal architectures.

In the context of next generation terminals, a multitude of radio access networks coexist and deliver different QoS according to used location, mobility, and applications. Interoperability and mobility management across these heterogeneous networks have become mandatory to enable full benefits of their complementary characteristics. This refers to concepts like Cognitive Radio (CR) and Software Defined Radio (SDR) introduced by Joseph Mitola [5]. The E^{2}R project [6] highlighted new services and reconfiguration mechanisms required in user equipment. Innovative services were proposed to identify appropriate Radio Access Technology (RAT) and to obtain expected QoS for running application. Moreover, specific mechanisms must also be supplied to dynamically adapt the radio interface to the best configuration. It consists in providing new services to manage, deploy, and configure RAT onto the platform resources. As an example, the Software Communication Architecture (SCA) has been proposed by the Joint Tactical Radio System (JTRS) program as a common infrastructure for efficiently managing software and hardware elements [7]. In this context, works described in [8] present a modeling approach based on the combined use of UML (Unified Modeling Language) and SystemC to simulate radio communication systems build upon SCA. In [9], an approach for the design of radio communication platforms is proposed. Modeled platforms are based on a specific architecture called NoTA (Network-on-Terminal Architecture). Functional and non-functional requirements are captured in UML2.0 and SystemC is used as the simulation language to evaluate performances of virtual architectures formed. However, this work does not address the modeling of adaptation mechanisms required for multi-standard terminals. A simulation framework for performance evaluation of SDR platforms is presented in [10]. The proposed approach makes possible to model baseband functions related to different protocol and to study their allocation on a SDR platform supporting simultaneously different RATs. Similarly to our approach, baseband functions are modeled at the packet level granularity and they are described only by their processing time. In this approach, the environment is related to a specific Medium Access Control layer (MAC) which limits the number of use-cases that can be simulated. The MOPCOM design methodology presented in [11] is proposed to enable design of SDR communication systems. It defines different levels of abstraction to correctly model baseband functions of radio communication systems and to allow automatic generation of VHDL and C/C++ codes. Nevertheless, even if the creation of SystemC models is claimed, performance evaluation of architectures is not addressed. A simulation-based approach is presented in [12] to carry out design space exploration of multi-core architecture on a UMTS data link layer design case study. Functional and architectural models defined separately are then mapped together to produce a system model with performance metrics. 48 mapping are investigated to estimate execution time of 11 tasks along with the average of 1 to 11 processing element utilization. Similarly to our approach, the purpose is to allow study of different architectures with light modeling effort. However, this work is limited by the number of operating scenarios considered.

The approach presented in this paper mainly differs from the above as to the way terminal architectures are modeled. In our approach, architecture specification is done graphically through a specific activity diagram notation. The behavior related to each elementary activity is captured as a finite-state machine to express the influence of application when executed on platform resources. The resulting architecture model is then automatically generated in SystemC to allow simulation and performance assessment.

Considering the modeling of flexible radio communication systems, specific methods are proposed to favor creation of efficient architecture models. In the following, the benefits of these methods are illustrated through a specific case study.

3. CONSIDERED MODELING APPROACH FOR PERFORMANCE EVALUATION OF SYSTEM ARCHITECTURES

Further information on the considered approach presented in this section is provided in [13]. In the considered approach, the architecture model captures information about both the structural description of the system application and the description of non-functional properties relevant to considered hardware and software resources. In order to provide efficient simulation speed and light modeling effort, architecture is modeled without considering a complete description of the application. Application is represented by a workload model. Workload models are used to represent the computation and communication loads applications cause on platform resources when executed. This approach is illustrated in Figure 1.

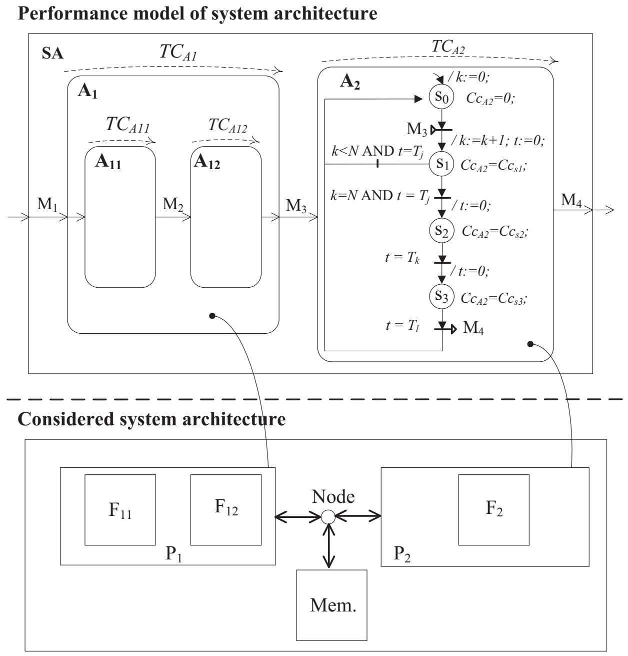

Figure 1 – Considered modeling approach for performance evaluation of system architectures.

The lower part of Figure 1 depicts a typical platform made of communication nodes, memories, and processing resources. Processing resources are classified as processors and dedicated hardware resources. In Figure 1, F_{11}, F_{12} and F_2 represent the functions of the system application. They are allocated on the processing resources P_1 and P_2 to form the system architecture. The upper part of the figure depicts the structural and the behavioral modeling of the system architecture. Each activity A_i represents a function, or a set of functions, allocated on a processing resource of the platform. As for example, activity A_{11} models the execution of function F_{11} on processor P_1. Relations M_i between activities correspond to transactions exchanged between activities. Transactions are exchanged through relations in conformity with the rendezvous protocol and they are defined as data transfer or synchronization between activities.

Behavior related to each elementary activity models the usage of resources by each function of the system application. Behavior exhibits waiting conditions on input transactions and production of output transactions. In the notation adopted, one important point is about the meaning of temporal dependencies. Here, transitions between states s_i are expressed as waiting transactions or logical conditions on internal data. A specific data value may be a time variable which evolves naturally. This data is denoted by t in Figure 1. The amount of processing and memory resources used is expressed according to the allocation of functions. In Figure 1, the use of processing resources due to the execution of function F_2 on P_2 is modeled by the evolution of the parameter denoted by Cc_{A2}. For example, Cc_{A2} can be defined as an analytical expression to give the number of operations related to the execution of function F_2. Value of Cc_{A2} can be influenced by data associated to the transaction received through relation M_3. In our approach, these properties could be provided by estimations, profiling existing codes, or source code analysis, as illustrated in [14]. The time properties used are directly influenced by the characteristics of the processing resources and by the characteristics of the communication nodes used for transaction exchange. Furthermore, the temporal behavior related to each activity is relevant to the function allocation. In case of a single processor, allocated functions are executed sequentially or according to a specific scheduling policy. In case of a multi-processor architecture, behaviors should express parallelism offered to execute functions.

Following this modeling approach, resulting model incorporates evolution of quantitative properties relevant to the use of processing resources, communication nodes, and memories. Using languages as SystemC, created architecture models can be simulated to evaluate the time evolution of performances obtained for a given set of stimuli. To facilitate creation of SystemC description we have used the framework CoFluent Studio [15]. This environment supports creation of transaction level models of system architectures. Graphical models captured and associated codes are automatically translated in a SystemC description. This description is executed to analyze models and to assess performances. Various platform configurations and function allocations can be compared considering different descriptions of activities. In Section 5, this approach is considered to model the SDR baseband architecture of an adaptive terminal.

4. TRANSACTION LEVEL MODELING OF FUNCTIONAL PROPERTIES OF AN ADAPTIVE TERMINAL

4.1. Considered case study

In this section we describe proposed case study to illustrate modeling approach for performance evaluation of SDR baseband architectures. This case study has been used to highlight modeling techniques proposed in [16] to capture new requirements identified in the next generation of adaptive multi-standard and multi-application radio communication system. The novelty in this paper lies in considering study of non functional properties of this system.

The considered case study reflects new features of these systems. It supports two different RATs and three user applications. An appropriate management of radio interfaces should maintain required QoS to ensure end-user experience in various radio environments. This new property implies to integrate new services within radio communication systems. The system should support dynamic activation/deactivation of one or more RATs. This adaptation mechanism should be monitored by a decision making module based on delivered QoS for each application. We present in Figure 2 the considered environment and relations with the radio communication system under study in the middle. The internal structure of such a system is refined using the activity diagram notation previously presented. In the considered case study, the user is supposed to request applications to the system through relations *Application request* and *Application response*. The system delivers three kinds of information related to the three applications supported: *Voice frame*, *Web page* and *Video frame*. Expected QoS for these applications are 20 frames per second for video, one frame of 160 samples every 20 ms for speech decoding and Web browsing corresponds to reception of varying size of data at about 1s. The network environment includes the activities related to application servers and transmission of data according to Universal Terrestrial Radio Access (UTRA) [17] and 802.11 [18] specifications. These radio links are depicted by three relations: *Downlink UTRA*, *Downlink WiFi*, and *Uplink UTRA*. The data size assigned per application in each radio frame depends on the channel quality of the link. For example, UTRA Network (UTRAN) is supposed to achieve at least 384 kbps in urban outdoor radio environment and at least 144 kbps in rural outdoor. WiFi has been defined to provide up to 54 Mbps data rate in indoor environment.

The internal structure of the system is composed of two main activities, *System management* and *System data flow* depicted in the upper and the lower part of Figure 2. Activity *System data flow* is made of activities concerned by data transmission in both uplink and downlink and application processing. UTRA and WiFi communication interfaces are depicted by *UTRA reception*, *WiFi reception* and *UTRA transmission* in activity *Radio reception and transmission*, whereas applications are depicted with activities *Voice processing*, *Web processing* and *Video processing*. The

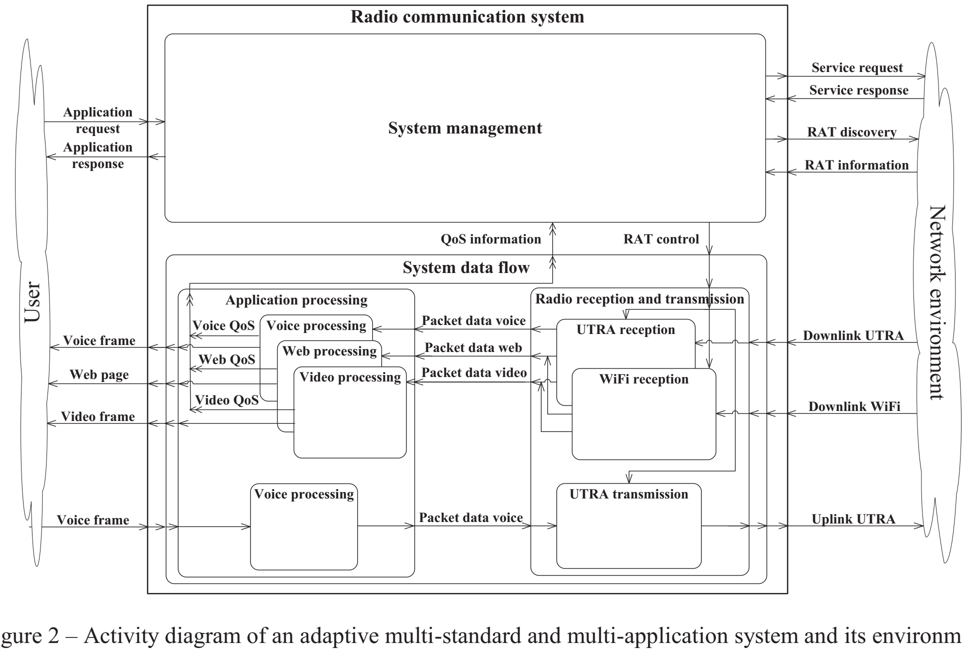

Figure 2 – Activity diagram of an adaptive multi-standard and multi-application system and its environment.

activities related to activity *System management* analyze QoS provided to user and perform request to the network environment. This activity receives information on data rate performance of the established radio link through the permanent relation *QoS Information*. In case of the QoS cannot be maintained, this activity supports a discovery service to find a RAT offering better performance. It is also able to decide and carry out a reconfiguration process of radio interface based on information provided by the *RAT control* relation. This process consists in activating the required communication interface. Besides, it communicates with the network environment according to the application request of the user and RAT to be used. Section 4.2 and 4.3 present proposed modeling techniques used to describe behavior of the system environment and communication interfaces.

4.2. Modeling the system environment

The behavior of the radio communication system is closely related to external behaviors of its environment. Indeed the system is supposed to adapt its radio interface according to the expected QoS level. This level is defined as the minimum data rate required by the user for each application. Therefore, it is necessary to model the system environment to evaluate multiple use cases and to study the dynamic behavior of the system. A generic solution consists in capturing the evolution of the environment as a scenario file containing the main time information, application declaration and associated parameters. A similar technique has been used in [9] to express sequence of service requests provided by functional requirements. Figure 3 illustrates the solution used to model the behavior of the user.

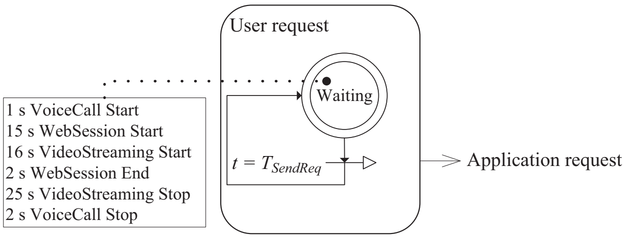

Figure 3 – Partial view of the activity diagram of the user.

The behavior related to the *User request* activity is defined to enable description of various usages of the multiple applications supported by the system. To achieve this, we propose to write in an user scenario file information describing the use case to evaluate. Information is organized as follows. Each line corresponds to an application request. The first field defines the time delay that must elapse before sending a request. The two other fields specify which application is inquired and if it must be started or stopped. This file is read in state *Waiting*. Time condition $t=T_{SendReq}$ expresses delay before sending transactions through relation *Application request*. $T_{SendReq}$ value evolves according to the time delay information indicated in each line. Transaction sent through relation *Application request* contains information related to the start or the stop of an application. The left part of Figure 3 gives an example of a user scenario file describing the successive use of the three applications proposed by the system under study.

We use a similar approach to describe the evolution of the network environment. Figure 4 presents a partial view of the activity diagram of the network environment.

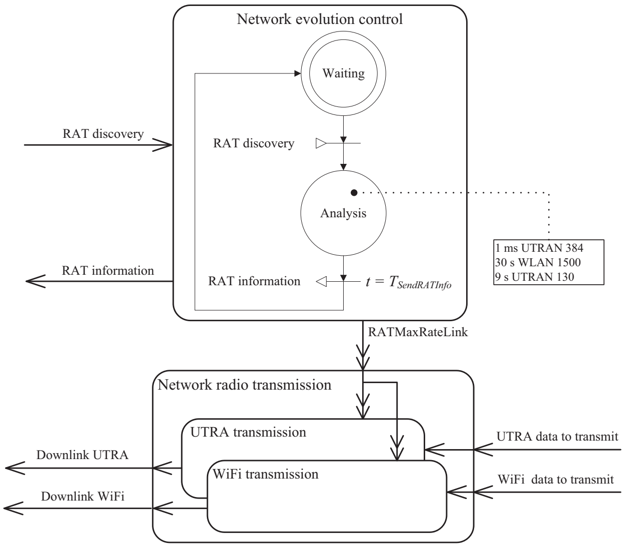

Figure 4 – Partial view of the activity diagram of the network environment.

The upper part of Figure 4 depicts the behavior of the *Network evolution control* activity. This activity is used to change the data rate offered by each RAT available in the network environment. We propose to write in a network scenario file information describing the evolution of the transmission conditions. Information is organized as follows. Each line of the scenario file defines one change in the transmission conditions associated to the network environment. A line is composed of three fields. The first one gives the time information when the transmission condition changes. The two next fields define which access network is affected among UTRAN and WLAN and what is the new data rate delivered. The state *Analysis* repeatedly reads one line in the file. After time delay, relation *RATMaxRateLink* is updating with the new data rate. Activities *UTRA transmission* and *WiFi transmission* use these information to define data rate could be used to send data on the radio links.

This approach based on scenario files makes possible to rapidly create various use cases. We can then easily assess different system operating scenarios for several combinations of the user and the network environment behaviors.

4.3. Modeling the communication interfaces

In this section, we present a modeling technique to capture communication interfaces at high abstraction level. Then we illustrate application of the proposed modeling approach to perform evaluation of the resource usage of the communication interfaces.

In our case, transactions identified in the data flow between the network environment and the radio communication system corresponds to the amount of data transmitted by each RAT. At this level, the associated payloads of transactions are defined according to expected throughput. For UTRA, transactions periodically occur every Time Transmission Interval (TTI) set to 10 ms and the maximum payload is 480 bytes. For WiFi, the maximum payload is 2347 bytes and the transaction instants depend of the size of data to transmit. Considering this data granularity, activities can be described as finite state machines. Furthermore, the described behavior is the same for the two RATs. Figure 5 illustrates the technique used to obtain description of the activities *UTRA reception* and *WiFi reception*.

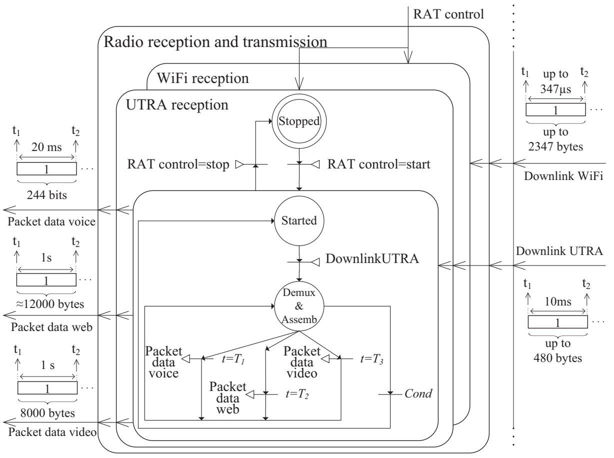

Figure 5 – Activity diagram of adaptive radio interfaces.

The behavior describes activities performed by layers involved in reception of data application (Medium Access Control layer, Radio Link Control layer for UTRA, Logical Link Control layer for WiFi). This type of description enables also to take into account the activation/deactivation of interfaces during system execution.

When *UTRA reception* or *WiFi reception* activities are started, information is exchanged with other activities in the form of transactions. As previously mentioned, transactions sent through relations *Downlink UTRA* and *Downlink WiFi* represent data transfer with the network environment. Transactions sent through the relations Packet data voice, Packet data web, and Packet data video represent the amount of data sent to related application processing activities. Packet data voice transaction must periodically occur every 20 ms with a payload of 244 bits. Packet data web transaction must be initiated 1s after the inquiring of the web page and contains quantity of data related to this web page. Packet data video transaction must periodically occur with a payload of 8000 bytes every 1s so as to display video at a rate of 20 frames/s. When *Downlink UTRA* or *Downlink WiFi* transaction is initiated by the environment, state *Demux & Assemb* in reception activities carry out both demultiplexing of data (voice, web, and video) and reassembling of previously segmented application data packets. When an application data packet is fully merged a transaction is initiated and the activity related to the application processing can start to deliver a service to the user. The behaviors of *UTRA reception* and *WiFi reception* activities evolve according to the *System management* activity. It decides to activate them through the relation RAT control. These two activities can be in two states: stopped or started. This description represents the dynamic behavior of the RAT interfaces at transaction level.

Modeling approach presented in Section 3 is applied to analyze performance obtained with different architectures to perform the chain of baseband processing related to physical layer of the two protocols. At the physical layer, synchronization, demodulation, channel equalization channel decoding, and multiple access channel extraction are linked together to form the chain of baseband processing. The behavior related to state *Demux&Assemb is* modified to allow an assessment of the related processing functions cost according architecture considered. Time interval between reception of *Downlink UTRA* or *Downlink WiFi* transactions and production of application data corresponds to time constraints to perform required baseband processing. The captured behavior enables to analyze the number of operations required to perform baseband processing related to each communication interfaces according to architecture considered.

4.4. Simulation results of the model

The CoFluent Studio tool has been used to generate and simulate an executable code of the model of our case study. It is obtained by the translation of a graphical model in a SystemC description. The code size related to this model corresponds to 4476 lines of SystemC code and 62% are automatically generated by the tool. In our case, simulation consists in analyzing how the system reacts according to various use cases. Functional performance evaluation enables to verify if properties such as timing constraints and QoS can be maintained by the system. The selected use case

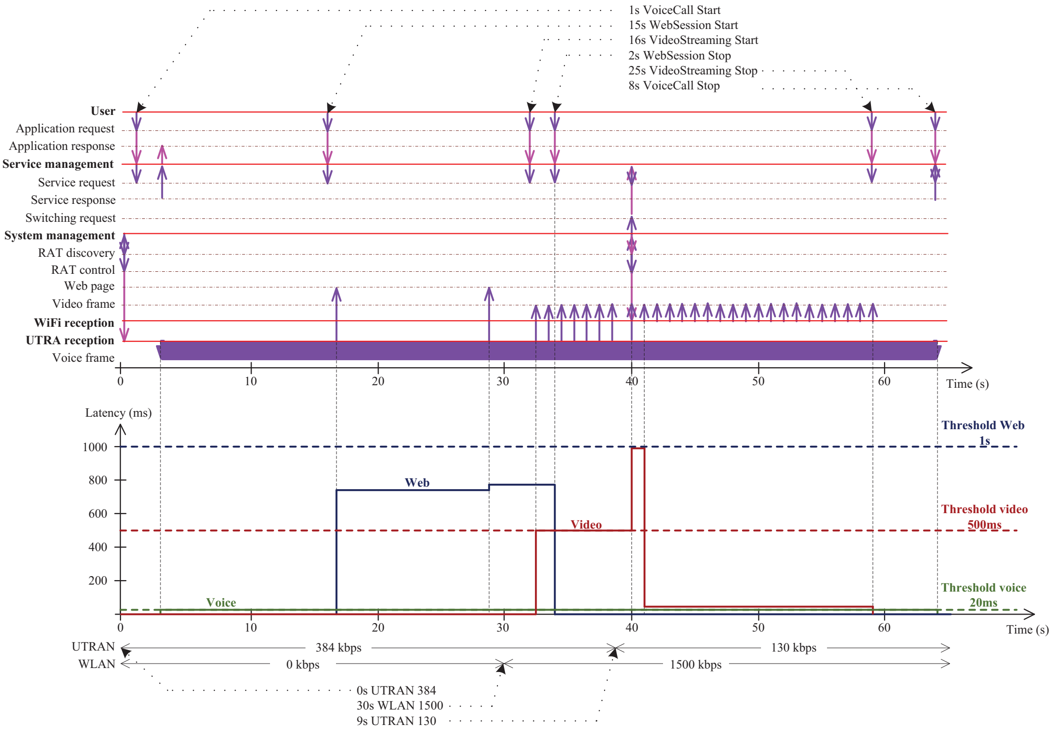

Figure 6 – Observation of transactions initiated during model simulation and related QoS evolution.

considers a user who successively requests the three applications supported by the system. Moreover the user evolves in a changing radio environment. At the beginning of the system execution, only UTRAN is available with a basic data rate set to 384 kbps to ensure data transmission required by all running applications. Then the data rate of the UTRAN is decreased to 130 kbps to model a changing environment. Therefore the WLAN is detected to support the video streaming application with the QoS expected. Figure 6 shows a way to display simulation results to observe the detailed behavior of the TLM model of the system.

The upper part of Figure 6 presents transactions exchanged during the simulation between activities. Activities and relations between activities are depicted on the left part. The lower part of Figure 6 presents evolution of the latency related to the three applications considered. This result of simulation is used to analyze the level of quality of services proposed to the user. On the right part the values of the latency threshold for each application are depicted. One can notice on the upper part of Figure 7 the successive start and stop requests of each application as expressed in the user scenario file depicted in the upper part. Initially the three applications are delivered by UTRAN. It is depicted by transactions sent by activity *UTRA reception* through the relation *Web page*, *Video frame* and *Voice frame*. After 39 s, the user is in an area where the data rate delivered by UTRAN is not sufficient to ensure both the QoS for voice and video streaming applications. Indeed, we observe on the lower part of Figure 6 that the QoS threshold related to video application is exceeded. The system performs a discovery process and identifies a new RAT in its environment. Then it decides to activate the activity *WLANreception* and the video streaming application is switched on to this RAT. This type of display is useful to validate the functional and behavioral model of the system with timing information.

5. PERFORMANCE EVALUATION OF THE FLEXIBLE BASEBAND ARCHITECTURE

5.1. Considered architecture

In this section, we present application of the proposed modeling approach described in Section 3 to perform evaluation of the resource usage of the communication interfaces. Here, we focus on the architecture required to perform UTRA and WiFi baseband processing. Figure 7 shows the studied architecture and the activities *UTRA reception* and *WiFi reception* presented in Figure 2.

The lower part of Figure 7 depicts the studied architecture. This architecture consists in implementing UTRA and WiFi baseband functions as a set of dedicated hardware resources denoted by P_1. For clarity reason, different functions related to UTRA and WiFi baseband processing are not represented. The upper part of Figure 7 depicts activities *WiFi reception* and *UTRA reception*. TC_{WiFi} and TC_{UTRA} denote the time constraints to be met by the related activities. TC_{WiFi} and TC_{UTRA} express the time constraints to satisfy when baseband functions are executed by P_1 each time a UTRA or WiFi frame is received. The behavior of these activities depicted in Figure 5 is modified in order to express the computational complexity per time unit each function causes on the resources when executed. Simulation results of the model makes possible to observe evolution of the usage of processing resources used to implement each baseband functions according to different operating scenarios. The computational complexity per time unit metric is considered here because it directly impacts area and energy consumption of the terminal architecture.

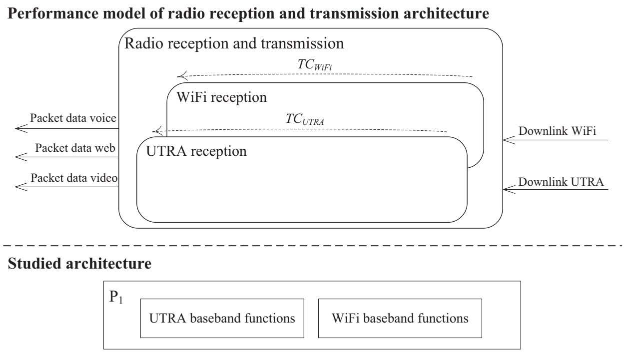

Figure 7 – Studied architecture to perform baseband processing related to activities UTRA reception and WiFi reception.

In the next section, we illustrate the simulation results could be obtained with our approach for one specific baseband function. We focus here on estimation of the required computational complexity per time unit to perform processing related to channel decoding function. Channel decoding functions for UTRA and WiFi are respectively performed with Turbo [19] and Viterbi decoder [20]. Computation duration and number of arithmetic operations are the two parameters required to estimate the resulting computational complexity per time unit. Based on a detailed analysis of resources required for channel decoding of UTRA [21] and WiFi [22], we have defined analytical expressions to give relations between functional parameters related to the different configurations of UTRA and WiFi frames and the resulting computational complexity in terms of arithmetic operations. The computation durations to perform channel decoding, denoted by $T_{ProcTurbo}$ and $T_{ProcConvol}$, are set in order to meet TC_{UTRA} and TC_{WiFi} time constraints.

5.2. Simulation results of the model

In this section, we present simulation results obtained to estimate maximal computational complexity per time unit required to perform UTRA and WiFi channel decoding with the considered architecture. Simulation of the model of flexible terminal enables to analyze the processing resource usage according to dynamic and non deterministic behavior of the system and its environment. Results presented are obtained considering the operating scenario described in Section 4.4. $T_{ProcTurbo}$ and $T_{ProcConvol}$ has been set to 2,5 ms and 4 ms to satisfy TC_{UTRA} and TC_{WiFi} time constraints.

Figure 8 presents a partial view of the evolution of the computational complexity per time unit related to the turbo decoding of data received by activity *UTRA reception*.

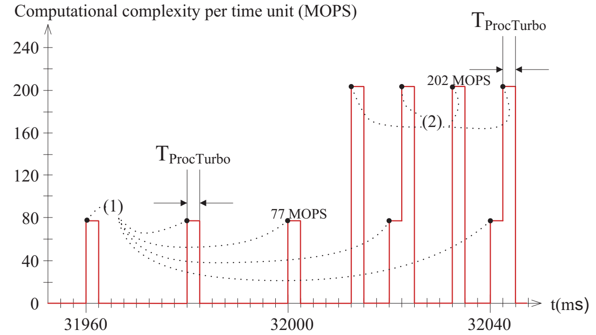

Figure 8 – Partial view of the time evolution of the computational complexity per time unit (in MOPS) for UTRA channel decoding.

Instants (1) and (2) correspond respectively to start instants of the channel decoding of data packets related to voice and video applications. We can observe resource utilization of the turbo decoder with the computation duration set. During this time interval the computational complexity per time unit varies between 77 MOPS and 202 MOPS.

Figure 9 shows the observation obtained with the simulation of the scenario presented previously.

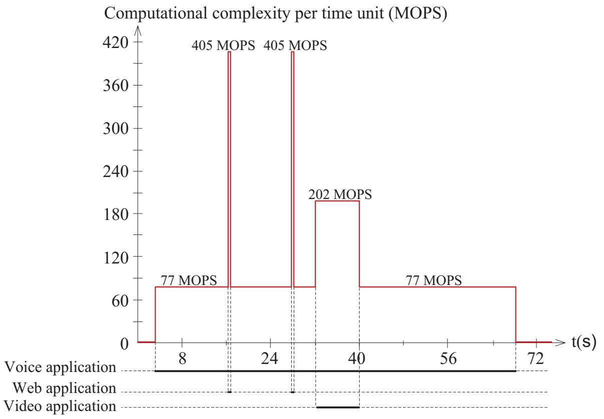

Figure 9 – Time evolution of the computational complexity per time unit (in MOPS) for UTRA channel decoding.

The lower part of Figure 9 depicts the resource utilization of P_1 according to the user applications supported. The upper part of Figure 9 shows evolution of the computational complexity per time unit for the UTRA channel decoding. We observe that a computational complexity per time unit of 405 MOPS is required for turbo decoding of Web data packets. This value corresponds to the maximum computational complexity per time unit that should be achieved by the hardware architecture to perform UTRA channel decoding with this operating scenario.

Figure 10 shows the observation obtained with the complete simulation of WiFi channel decoder considering the scenario previously presented.

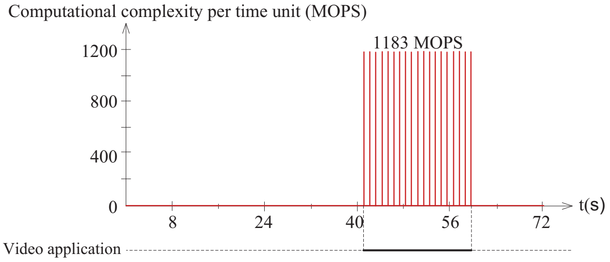

Figure 10 – Time evolution of the computational complexity per time unit (in MOPS) for WiFi channel decoding.

In this operating scenario, the WiFi reception interface is used to receive video streaming data. This radio interface is activated after 40s. We can observe that a maximal computational complexity per time unit of 1183 MOPS is also required. To estimate expected resources for this architecture, it is necessary to analyze the global computational complexity required to execute simultaneously UTRA and WiFi channel decoding.

Figure 11 presents the partial view of the global computational complexity per time unit for studied architecture with operating scenario considered.

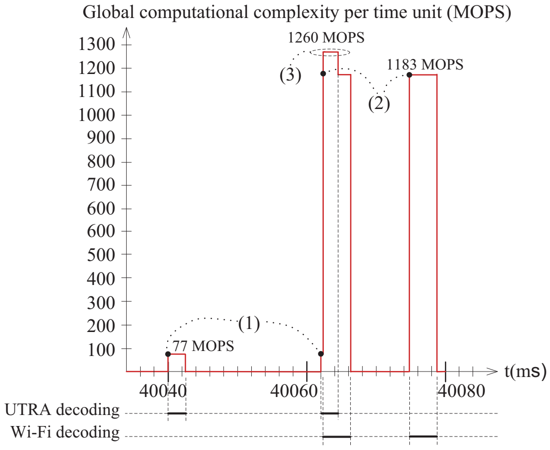

Figure 11 – Time evolution of the global computational complexity per time unit (in MOPS) for UTRA and WiFi channel decoding.

The lower part of Figure 11 depicts the resource utilization of P_1 to perform both UTRA and WiFi channel decoding. The upper part of Figure 11 shows the selected time interval where the maximal computational complexity per time unit is observed. Instants (1) and (2) correspond respectively to start instants of the channel decoding of data packets related to UTRA and WiFi protocols. We can observe resource utilization of the turbo decoder with the computation duration set. The time interval denoted (3) shows the maximal computational complexity per time unit observed with this operating scenario and architecture considered. It is estimated to 1260 MOPS. It occurs when UTRA and WiFi channel decoding are performed simultaneously. This kind of result highlights interest in simulation based approach compared to analytical approach. Indeed, the resource usage evolves dynamically because different wireless protocols with different range of performance requirements can be switched at unknown timing moments. The simulation time to execute the model for the operating scenario considered took about 500 ms on a 2.66 GHz Intel Core2 duo machine. Simulation time achieved and proposed approach based on scenario files enable system architect to iterate quickly over different operating scenarios to evaluate and compare resource usage. These simulation results can be used to study different resource utilization according to different RAT operating modes. Moreover, proposed modeling approach allows quick performance comparison with other potential architectures.

6. CONCLUSION

Radio communication systems are assumed to support new services to provide adaptation capabilities of radio functionalities according to internal and external changes. One important challenge is to provide performances evaluation of radio communication architecture highlighting these new requirements. In this paper, we have presented a simulation-based approach and modeling techniques to evaluate by simulation performances of an adaptive multi-application and multi-standard system. This approach leads to the creation of abstract representation of the architecture to efficiently deliver simulation results. Finally, the related computational complexity required to perform baseband signal processing for the various RAT supported has been evaluated to make possible platform sizing. Further research is directed towards validation of estimates provided by simulation and applying the same modeling principle to other non-functional properties such as dynamic power consumption.

10. REFERENCES

[1] J. Govil, “4G Mobile communication systems: turns, trends and transition”, *In Proceedings of the International Conference on Convergence Information Technology (ICCIT’07)*, Gyeongju, Republic of Korea, November 2007.

[2] K. van Berkel, “Multi-core for mobile phones”, *In Proceedings of Design, Automation and Test in Europe (DATE’09)*, 2009.

[3] D. Densmore, R. Passerone, and A. Sangiovanni-Vincentelli, “A platform-based taxonomy for ESL design”, *IEEE Design and Test of Computers*, vol. 23, no. 5, pp. 359-374, September/October 2006.

[4] M. Gries, “Methods for evaluating and covering the design space during early design development”, *Integration, the VLSI Journal*, vol. 38, no. 2, pp. 131-183, 2004.

[5] J. Mitola III, “Cognitive radio: an integrated agent architecture for software defined radio”. Ph.D. dissertation, Computer Communication System Laboratory, Department of Teleinformatics, Royal Institute of Technology, Stockholm, Sweden, May 2000.

[6] End-to-End Reconfigurability. (2007) IST project summary. ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/cnt/e2rii_en.pdf

[7] J. Bard and V. J. Kovarik, *Software defined radio: the software communication architecture*, John Wiley & Sons, March 2007.

[8] G. Gailliard, E. Nicollet, M. Sarlotte and F. Verdier, “Transaction level modeling of SCA compliant software defined radio waveforms and platforms PIM/PSM”, *In proceedings of the Design, Automation and Test in Europe Conference (DATE’07)*, San Diego, California, April 2007.

[9] K. Kronlöf, S. Kontinen, I. Oliver and T. Eriksson, “A method for mobile terminal platform architecture development”, I*n the proceedings of the Forum on specification and Design Languages (FDL’06)*, Darmstadt, Germany, September 2006.

[10] M. Trautmann, and al., "Simulation Framework for Early Phase Exploration of SDR Platforms: A Case Study of Platform dimensionning," *in proceedings of Design, Automation and Test in Europe Conference (DATE'09)*, Nice, 2009, pp. 312-316.

[11] S. Lecomte, S. Guillouard, C. Moy, P. Leray, P. Soulard. “A co-design methodology based on model driven architecture for real time embedded systems”, *Mathematical and Computer Modelling*, vol. 53, no. 3-4, pp. 471-484, 2011.

[12] D. Densmore, A Simalatsar, A. Davare, R. Passerone and A. Sangiovanni-Vincentelli, "UMTS MPSoC Design Evaluation Using a System Level Design Framework," *in proceedings of Design, Automation and Test in Europe Conference (DATE'09)*, Nice, 2009, pp. 478-484.

[13] A. Barreteau, S. Le Nours and O. Pasquier, “A State-Based Modeling Approach for Efficient Performance Evaluation of Embedded System Architecture at Transaction Level” *Journal of Electrical and Computer Engineering*, vol. 2012, Article ID 537327, 16 pages, 2012.

[14] J. Kreku, M. Hoppari, T. Kestilä, and al., “Combining UML2 application and SystemC platform modelling for performance evaluation of real-time embedded systems”, *EURASIP Journal on Embedded Systems*, vol. 2008, 2008.

[15] CoFluent Design, http://www.cofluentdesign.com/

[16] A. Barreteau, S. Le Nours and O. Pasquier, “Executable models for performance assessments of adaptive mobile systems”, I*n the proceedings of Software Defined Radio Technical Conference (SDR’09)*, Washington DC, USA, 2009.

[17] 3GPP. 3GPP Technical Specification Group, 25 series: Radio Aspects. Technical specification (Release 8).

[18] IEEE Standard association. IEEE 802.11-2007, IEEE standard for information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Technical specification, June 2007.

[19] 3GPP. (2002) TS 125.201, UMTS; Physical layer - general description. http://www.3gpp.org/ftp/Specs/html-info/25201.htm

[20] IEEE Computer Society. (2007) IEEE Standard association. IEEE 802.11-2007, IEEE standard for information technology - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. [Online]. http://www.ahltek.com/WhitePaperspdf/802.11-20%20specs/802.11-2007.pdf

[21] 3GPP. (2002, Sep.) TS 125.212, UMTS; Multiplexing and channel coding. [Online]. http://www.3gpp.org/ftp/Specs/html-info/25212.htm

[22] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.