REAL-TIME SCHEDULING ANALYSIS FOR DSP BASE BAND PROCESSING IN MULTI-CHANNEL SDR EQUIPMENTS

Noël Tchidjo Moyo[1, 2], Eric Nicollet[1], Frederic Lafaye[1], Christophe Moy[2]

[1]THALES Communications, Colombes, France
[2]SUPELEC/IETR, Campus de Rennes, France
noel.tchidjomoyo@fr.thalesgroup.com, eric.nicollet@fr.thalesgroup.com,
frederic.lafaye@fr.thalesgroup.com, christophe.moy@supelec.fr

ABSTRACT

In perspectives offered by multi-channel SDR set, digital signal processors (DSP) need to simultaneously execute several radio physical layer (PHY) components. One major challenge in integrating together such DSP applications is the real time requirements that must be satisfied. In order to satisfy these real time requirements, specific needs of real time scheduling analysis arise because of simultaneous executions on the DSP platform.
In this paper, we propose a design approach, which allows the real time scheduling analysis of tasks on DSP. A real-time scheduling simulation tool is also presented to help SDR applications designers; it is able to test the feasibility of tasks whose execution times, deadlines, and minimum separation times are different from one phase to another. The tool is also able to verify (by simulation) the real-time behavior of tasks implemented in DSP and scheduled according to dynamic priority scheduling. We also discuss the impact of the study on the capabilities of SDR equipments.

1. INTRODUCTION

The development of several communication standards on the same piece of hardware highly increase the need for testing the feasibility (in terms of real time scheduling) of the application, when implementing the new standard or update an existing one.
Designers of such SDR applications have resort to ad hoc methods and perform pre-runtime real-time scheduling on DSP by hand. Since the complexity, due to the different combinatorial of waveforms, keeps growing, ad hoc and manual methods will prone to errors, time consuming and will often fail to find a feasible schedule even when one exists.
Indeed waveform applications, like other real time systems, have a dual notion of correctness: logical and temporal. It is not sufficient to only produce the correct outputs; the correct outputs should be produced within the correct time interval. We propose to integrate the analysis in a model driven architecture (MDA) approach. In a first phase, the functional aspects of the waveform applications are specified with their times constraints.
In a second phase the platform architecture is specified in terms of hardware capabilities, real-time operating systems to be used etc.
In a last phase, functional specification is mapped on to platform architecture, and then the tool performs the scheduling analysis.

Too often, current tools are aiming at directly providing code generation after a very limited modeling, and do not take into account real time scheduling analysis. On the other hand it is important for the designers of SDR applications to know as soon as possible the real time scheduling of waveform applications. This allows undertaking early on the correct implementation solutions.

The remainder of this paper is organized as follows. In the next section backgrounds and related work are presented. Section 3 describes our design approach proposal. Real-time scheduling issues are discussed in section 4. Section 5 provided MDE design environment. The impact of the study on the capabilities of SDR equipments is provided in section 6. Conclusions are provided in section 7.

2. BACKGROUNDS AND RELATED WORK

Much work has been carried out to model SDR applications, in a SCA context in particular. Zeligsoft CE tool developed by the company Zeligsoft and Spectra tool developed by Primstech allow code generation for software components in software radio application [1] [2]. These works do not offer capabilities to investigate real-time constraints for real-time scheduling.
Concerning MDA approaches for SDR design, the UML profile A3S [9] was a first attempt for the design of SDR systems. A few verification means were also addressed in this research work.
MDA modeling for real-time embedded equipments such as SDR is also addressed in current research project MOPCOM. MOPCOM aims at automatically generating simulation code (SystemC for instance) and implementation

code (CatapultC or VHDL) based on a MDA design methodology [10]. It uses in particular MARTE profile.

The UML profile MARTE [3] has been standardized at the OMG, to provide real time extension for specification and design of embedded systems. Thus a plug-in has been developed for the modeling tool Rational Software Architecture to integrate MARTE profile; hence a bridge has been developed to export the model to the scheduling analysis tool Cheddar developed by the University of Brest [4]. Another bridge has been developed to export the model for the scheduling analysis tool Rapid RMA developed by the company Tri Pacific Software. These last approaches are too generic and not consider, modular construction and reuse of signal processing functions, technical services such as connectivity, operating systems, time management services etc.

3. A DESIGN APPROACH PROPOSAL

3.1 SDR design context

Let us consider two radio waveforms (possibly n) that must be executed on several hardware platforms. Radio processing is nothing else than some signal processing, so that it can be done by a processor instead of ASICs as done in the past, as soon as the processor is powerful enough to execute the signal processing in real-time. Let us be interested in the execution feasibility on a processor of interest.

3.2 PIM modelling

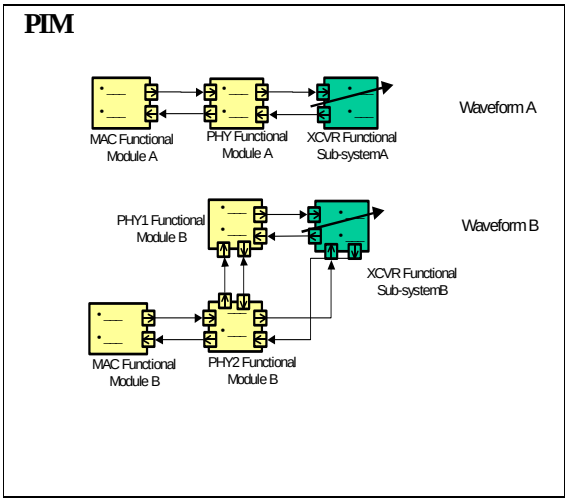

Figure 1 – PIM top view model for two radio waveforms

In a MDA design approach, the first design phase consists in realizing a model independent from any platform: PIM (platform independent model). Figure 1 represents a PIM view of two radio waveforms, one with three modules and the other with four modules.
A functional module belonging to the MAC (Medium Access Control) layer, a functional module belonging to the physical layer and a transceiver sub-system, compose the "waveform A".
A functional module belonging to the MAC layer, two functional modules belonging to the physical layer, and a transceiver sub-system, compose the "waveform B". We can also see on this PIM the various connections between the functional modules. A functional module is defined as a module taking part to a waveform functional decomposition and participating to the realization of the concerned waveform capability.

The PIM represents only the business functional capacities (here radio) and the behavior of the system, without deterioration due to technological considerations. The clarity of this model allows experts of the domain to understand much better than an implementation model. They can verify more easily that the PIM is complete and correct.

Once the PIM is detailed enough, the second stage consists in mapping this model towards a specific platform: PSM (platform specific model). To obtain a specific model, it is necessary to choose one or several execution platforms (several platforms can be compared to implement the same model).

3.3 PSM modelling

The transceiver sub-system of the PIM model is transformed into two parts at PSM level as shown in Figure 2. The first one, which serves as façade, allows from DSP, to pilot the second part, which corresponds to the core of the transceiver and is situated on a FPGA. Façades are parts of the functional environment.

Inside a given execution unit, the functional environment is the platform support software specifically present in the execution unit to provide functional support. Functional support completes the waveform with the processing on radio domain contributing directly to the implementation of waveform capability.
In the PSM model appears new software services offered by the platform. The reconfiguration infrastructure is defined as the software sub-system of the radio set which is undertaking the management of a reconfiguration process, ensuring the transition towards the physical configuration state decided by decision making.
From components model point of view, inner connectivity represents connectivity solution between software components residing in the same address space.

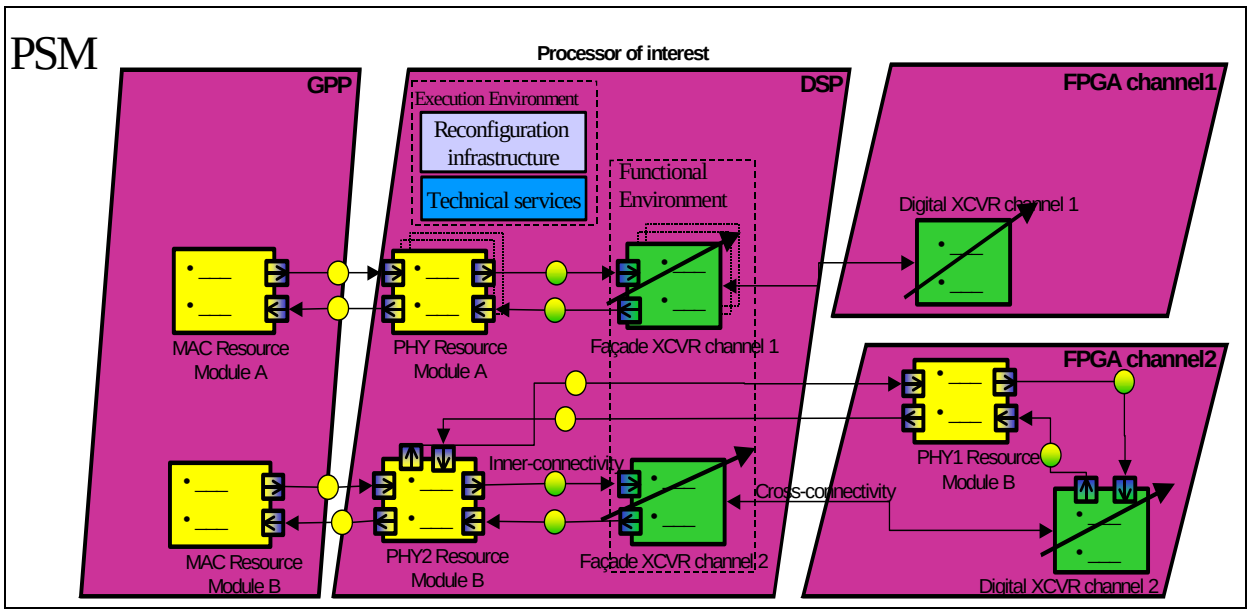

Figure 2 – PSM top view model of the two radio waveforms

Cross connectivity represents connectivity solution between software components residing in different address spaces. These software services can be grouped together to set called "execution environment"(EE).

3.4 From modelling to real-time scheduling

The PIM model can integrate architectural and temporal aspects but always without specific details depending on an implementation platform. These models can contain for example deadlines, which should be met in transmitting and receiving steps. At the PIM level one "functional configuration state" can be selected. A functional configuration state is characterizing a state that the platform may reconfigure into.
Then, one "physical configuration state" must be made. A physical configuration state is defined as the mapping of functional modules on the various execution units.

The projection on platform of the PIM models of waveforms is made with the addition of characteristics of software services supplied by the platform (times transfer due to the connectivity used etc.), plus the worst-case execution time of the different functional modules.

Hence, we propose to determine the real time scheduling of tasks on the DSP processor.

4. REAL-TIME SCHEDULING ISSUE

4.1. Context

Consider that the waveform A corresponds to a multi-user implementation of a modulation scheme. The SDR application can serve multiple users/services through different frames size handling. Video frame usually takes much more processing time than the voice frame, that implies for a task of the DSP processor to have different execution times, deadlines and minimum separation times, depending on the type of the frame. Typically, the PHY resource module A (of figure 2) is composed of a flexible decoding function for which different execution times are due to different parameters for audio and video frame.

The closest task model from real-time scheduling literature, is the generalized multiframe (GMF) task model [5], where execution time, deadline and minimum separation time are changed periodically according to a specify pattern.

The main difference between the GMF task model and the task model we face in our design context comes from the fact that there is no activation pattern. This is because frames sequences are random in both directions (transmitting and receiving sides) and we cannot presuppose an arrival sequence (user A makes an audio call while user B watches a video broadcast). Thus known algorithms, proposed for the feasibility test of GMF tasks cannot be applied, since there is no cycle. We call this new task model: non-cyclic GMF task.

Figure 3 presents two tasks models. In the top one is the GMF task model for which a scheduling feasibility test exists. The bottom scenario is the one we consider in this paper. The difference is that the succession of frames does not follow a pattern.

C_i^j , D_i^j , P_i^j represent the maximum execution time of the j-th frame of the task i, the relative deadline of the frame, and the minimum separation time between the frame and the following one, respectively.

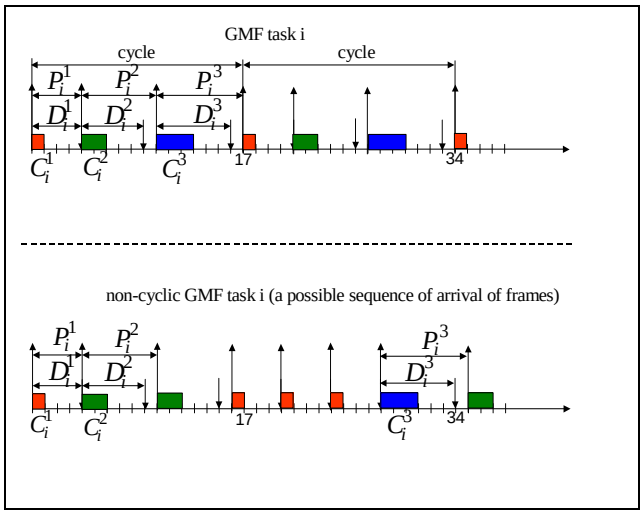

Figure 3 – Execution comparison between GMF task i and non-cyclic GMF task i

In the following section we discuss the schedulability of the non-cyclic GMF tasks set.

4.2. Feasibility test

The analysis on this paper considers *m* independent non-cyclic GMF tasks executing on a uniprocessor according to the preemptive earliest deadline first (EDF) scheduling policy. We consider that the tasks satisfy the *Frame Separation* (FS) property [6]. A GMF task is defined to satisfy the FS property, when the absolute deadline of each frame is no later than the arrival time of the following frame.

We have been deriving the following formula from our study:

$$\sum_{i=0}^{m-1} \max_{0 \le j \le N-1} \left(\frac{C_i^j}{D_i^j}\right) \le 1$$

When this formula is true then the non-cyclic GMF task set is schedulable under EDF scheduling. The idea behind this formula is that, if we can schedule a set of tasks when taking for each task the scenario where the task deals with the frame, which generates the maximal criticality ($\frac{C_i^j}{D_i^j}$), then the set of tasks will be schedulable whatever the arrival sequence of frame is. Please note that this is a sufficient condition, when the condition is not satisfied, no answer can be given concerning the schedulability of non-cyclic GMF tasks. To solve this last issue we provide a real-time scheduling simulation tool

4.3 Real-time scheduling simulation tool for event-driven SDR system

The tool is based on finite state automata. The main idea is to associate each state of the automata to the state of the processor (for example: idle, run, deadline miss), and a transition to either the arrival of a frame or a situation in the ready queue.

A random frame list is generated and distributed in the time. There is a clock representing the time progression. Whenever a task is released due to the presence of data to process, it will be put in a ready queue to be executed.
When the clock steps, we decrease the execution time of the task in the first position of the ready queue, and the deadline of all tasks in the queue. A task instance will be removed from the queue when its execution time becomes 0. The automata will reach the state “deadline miss” when the execution time of a task instance in the ready queue becomes greater than its relative deadline. The state “idle” is reached when the ready queue is empty.

The Figure 4 illustrates a model of such automata.

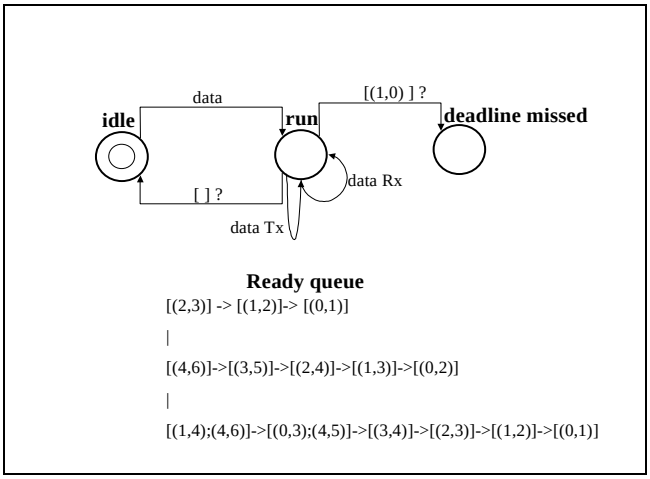

Figure 4 – A model of automata scheduler with ready queue

In the following section, we describe the overall design environment used in our case study for the demonstration.

5. MDE DESIGN ENVIRONMENT

The UML modeler tool used is Rational Systems Developer with the UML profile MARTE.

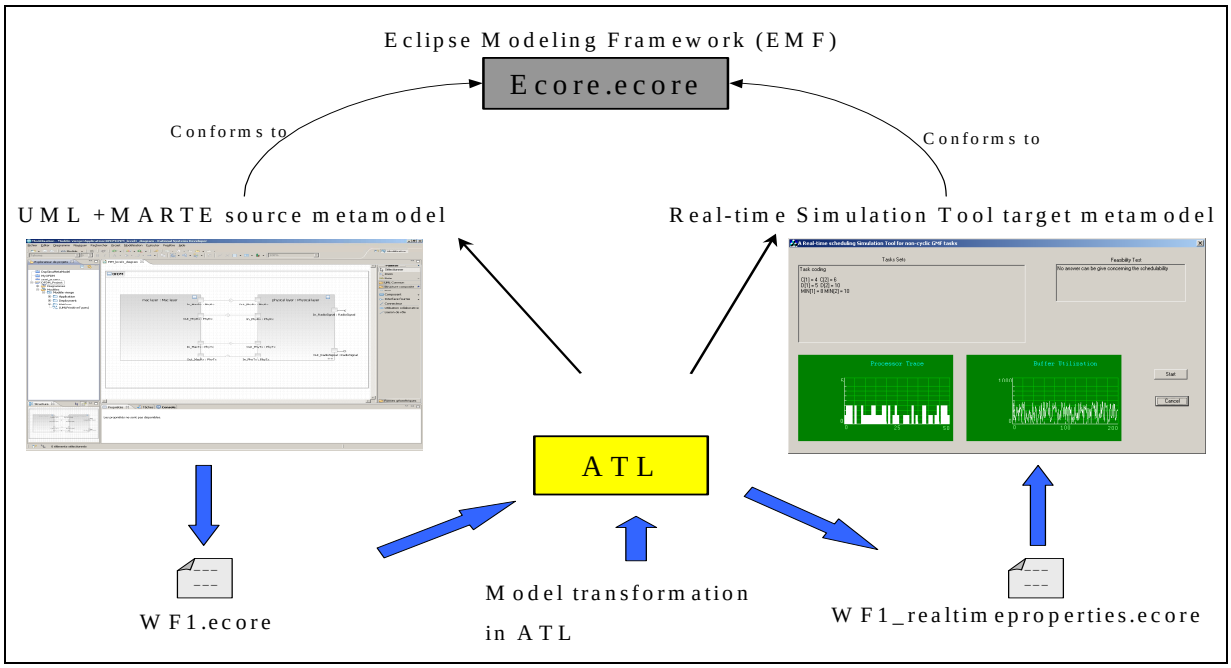

Figure 5 - MARTE to Real time Simulation tool transformation using ATL

5.1. MARTE profile

The UML profile MARTE replaces the former Scheduling Performance and Time (SPT) profile [3]. It is OMG compliant with other standards in the domain, such as SysML, SAE AADL, EAST-ADL2 and ARINC 653. The profile implementation for Rational Systems Developer is currently available in open source.
MARTE profile implementation allows expressing real-time properties on the models. We matched MARTE concepts to the real-time scheduling simulation tool concepts, and used Atlas Transformation Language (ATL)[7] in order to transform MARTE model to the real-time simulation tool model.

5.2 MARTE to real-time simulation tool transformation using ATL

ATL is a model transformation language. In the field of Model-Driven Engineering, ATL provides ways to produce a set of target models from a set of source models. In order to achieve the transformation, as shown in figure 5, we need to provide:

- UML and MARTE source metamodel
- Real-time simulation tool target metamodel
- A waveform model
- A transformation model in ATL

When the ATL transformation is executed, we obtain a target model conforming to the real-time simulation tool metamodel. The tool to perform the analysis can read this target model.

5.3 Project organization

As shown in figure 6, the project is composed of three packages: application, deployment and platform. In the application package, there are different levels of the various waveforms PIM. These waveforms PIM are represented in the form of class diagrams. For each waveform we have six levels of PIM. Then, in the platform package, there is a diagram, which describes the architecture of the platform. Finally, in the deployment package, there is a diagram, which represents the mapping of a set of waveforms on the various execution units.

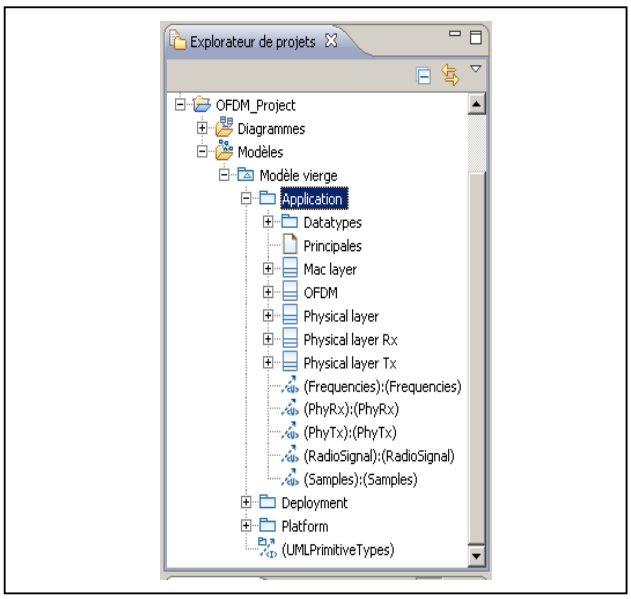

Figure 6 - Project explorer

5.4. The results of the real-time scheduling tool

The tool first verifies if the schedulability condition presented in section 4.2 is respected. If it is not the case a real-time simulation is launched, based on a large random frames list. The designer can watch the different states of the processor, and at the same time the buffer utilization (please note that we associate to each task a buffer requirement).
When a deadline is missed the processor passes through the state "deadline missed", the simulation stops and a log file contains the sequence of frames which brought to miss a deadline, with the tasks present on the ready queue.
The following section discusses about the impact of the study on the capabilities of SDR equipments.

6. IMPACT OF THE STUDY ON THE CAPABILITIES OF SDR EQUIPMENTS

The EDF dynamic scheduling policy, chosen in this study, allows more flexibility in reconfiguration use cases. Indeed, after verifying the feasibility (in terms of real-time scheduling) of the state we want to reconfigure into, no changes need to be made on previous tasks of the physical layer on the DSP processor. This is because in EDF scheduling, tasks are placed in a queue and whenever a scheduling event occurs (tasks finishes, new task released, etc.) the queue will be searched for the process closest to its deadline. Hence, we do not have to modify priority values of previous tasks, as it would have been if we had used fixed priority scheduling.

The real-time scheduling simulation tool ensures that the new standard to be implemented on the platform satisfies real-time requirements. It then increases the reliability of SDR equipments. The buffer utilization represented in the scheduling tool, allows a better management of memory usage as well. This contributes to increase performance of SDR equipments.

7. CONCLUSIONS AND FUTURE WORK

An important step in the development of embedded real-time SDR systems is "schedulability analysis" on DSP processor. The goal is to check whether all tasks can be executed within the given deadlines in all possible scenarios. In this paper, we present a design approach, which allows determining the real-time scheduling of tasks, implementing several physical layers components and executing simultaneously on a DSP processor. The real-time scheduling simulation tool presented in this paper provides useful analysis (feasibility test, processor trace, buffer utilization) to the designer of event-driven SDR system. This minimal time spent to ensure the feasibility of the implementation at the very beginning of the design phase, is a valuable time saving in a design process that requires months. This ensures the designer that the way in which he engages will meet his temporal requirements in the final product. As future work, we plan to extend the analysis to DSP multi-core processor.

REFERENCES

[1] http://www.zeligsoft.com
[2] http//www.prismtechnologies.com
[3] S. Gerard, D. Petriu and J. Medina, "Marte: a new standard for modelling and analysis of real time and embedded systems", Pisa, Italy, July 3rd, 2007.

[4] http://www.omgmarte.org/Tools.htm

[5] S. Baruah, D. Chen, S. Gorinsky, A. Mok. Generalized multiframe tasks. 1996

[6] H. Takada, K. Sakamura. Schedulability of generalized multiframe task sets under static priority assignment. *In Real-time Computing Systems and Applications*, pages 80-86, 1997.

[7] http://www.eclipse.org/m2m/atl/

[8] C. Norstrom, A. Wall, W. Yi. Timed automata as tasks models for event-driven systems. *In Real-time computing and applications*, pages: 182-189 1999.

[9] S. Rouxel, J. Diguet, N. Bulteau, J. Carre-gourdin, J. goubard, C. Moy. UML Framework for PIM and PSM Verification of SDR systems. *SDR Forum Technical Conference'05*, Anaheim (USA), November 2005.

[10] Stephane Lecomte, Christophe Moy, S. Guillouard. A Co-Design methodology based on model driven engineering for SDR equipments. *SDR Forum Technical Conference'09*, Washington DC (USA),1-4 December 2009.