COGNITIVE RADIO EQUIPMENTS MODELING: META-MODEL AND TOOLING PROPOSALS

Oussama LAZRAK, Christophe MOY, Pierre LERAY

SUPELEC/IETR, Cesson-Sévigné, France
christophe.moy@supelec.fr

ABSTRACT

A Model Driven Architecture (MDA) approach is proposed in this paper for the design of Cognitive Radio (CR) equipments. MDA consists in describing a system at different levels of abstraction, from a high level abstraction representation (often graphical) downto the code to be executed into a specific platform (made of pieces of hardware - e.g. executed on ASICs or FPGAs - and software - e.g. executed on DSPs or GPPs -). This approach for CR is equivalent to define a universal language that any designer in the community may understand in order to design CR devices. Associated to this language, a design environment is proposed in order to provide the adequate tools to design the models at each level of abstraction relatively to CR design needs. This work is based on our previous work on HDCRAM.

1. INTRODUCTION

Radio equipments' design is getting more and more complex. But it is true for many new systems. The reason is that a radio equipment is no more a single purpose device but also combines several processing devices, states and operation modes. Everyone has an experience in using a wireless phone today. If this multiplicity is obvious at the application level, users may not be aware that it is also true at the radio processing level. Radio processing is done by a set of electronic circuits, the operations of which are not only wired and hardly pre-defined. They are more and more managed and configured by some software control. This implies a design complexity in terms of span of possible use cases, and makes it more difficult to guaranty a perfect operation in any context.

However, current state is just the premise of a future generalization of software and hardware mix. We usually refer to Cognitive Radio (CR) when speaking about future highly flexible and auto-adaptable radio devices [1][2]. Cognitive Radio is a general statement to speak about radio devices that would adapt in real-time to any aspect of their environment, so that they can perfectly match their operation to the current environment state and in function of the current needs [2]. As an example of a typical CR scenario, we may think at saving the batteries while adapting the processing tasks in real-time to the propagation constraints or the battery level of a radio device. However, most of the time, CR refers to new spectrum usages, such as dynamic spectrum access (DSA) and opportunistic spectrum access (OSA) [3][4].

Anyway, current and future radio devices require new paradigm in the design phase. We assert this consists in decomposing the design in several steps, including new phases where some realities are abstracted at some levels, e.g. modeling, before producing the conventional implementation code. In that way, designers can manage the design complexity in a step by step basis that permits to mitigate programming errors and final product malfunctions. This paper is all about proposing an adequate modeling approach for CR equipments design. We argue that the specification of CR compared to conventional radio design is the management architecture that is essential to be added to the usual radio signal processing, so that the cognitive features and behaviors can be efficiently integrated in the radio system.

2. MDA APPROACH

Abstractions of the equipment should rely on established modeling means. MDA (Model Driven Architecture) is a standardized approach [5], based on UML (Unified Modeling Language), to describe a system composed of software and hardware pieces. A first step consists in modeling the application regardless of the platform of execution (PIM : Platform Independent Model), followed secondly by a description of the platform (PDM : Platform Description Model) that can be a software platform such as .NET, JAVA, or hardware like DSP, FPGA, GPP, etc. Then the PIM is translated to one or more Platform Specific Models (PSM) [6], which has been firstly explored for Software Defined Radio a decade before [7][8][9]. The development of automated tools, making it easier to deal with new technologies, and allowing the implementation of real systems is still a research area [10][11]. These Tools are mostly based on transformation tools of models [12], such as ATL (Atlas Transformation Language), BOTL (Basic Objet-Oriented Transformation Language), QVT or ACCELEO. MDA approach is related to multiple standards, including UML as already stated, and XML Metadata Interchange (XMI) for a general purpose modeling, or the Meta-Object Facility (MOF) to define its own standard on a specific domain [13].

We propose here a MDA design approach for Cognitive Radio, that enables to specify and design the management architecture that can support the real-time adaptation and reconfiguration of the radio processing, whatever the implementation (HW and/or SW). In particular, we proposed HDCRAM to specify how to include sensing [2] and decision making [14] (including learning) features into CR equipments. HDCRAM stands for Hierarchical and Distributed Cognitive Radio Architecture Management [15]. HDCRAM is a set of rules to respect in order to generate models compatible with the meta-model that we defined for CR. Note that these features are those of any auto-adaptive system. The proposed approach thus is not restricted to radio domain and can be applied to any auto-adaptive system design.

3. GENERAL-PURPOSE MODELING

To ensure the portability and the compatibility between radio devices, JTRS [16] defined an open architecture mainly expressed in UML. This architecture, called Software Communications Architecture (SCA [16]), enables the design and deployment of software and hardware components within an SCA-compliant system. The SCA specification is based on several UML diagrams, and uses the class diagram to express the components entities of radio equipments as well as their connections. On the dynamic point of view, UML provides other adequate diagrams, such as sequence diagram, which SCA uses to define the messages exchanged between these entities. Transition diagrams and Object diagrams could be a practical alternative. But these diagrams are not quite formal [17]; therefore they are rarely used for generating some code automatically. Hence there is a need to combine class diagrams with a formal language, in order to express the system evolution over time, which is possible by using OCL (Object Constraint Language). OCL however adds more complexity on PIM modeling, and also relies on specific constraints, without guarantying implementation success. This is contradictory to the MDA approach fundamentals [18] as MDA aims at improving the productivity gain by abstracting complexity and making possible the reuse of models. To solve these problems we suggest a modeling approach specific to CR, based on the HDCRAM meta-model.

4. COGNITIVE RADIO MODELING

4.1. HDCRAM architecture

CR equipment is made of a set of software and hardware components interconnected in order to perform the expected functions, e.g. radio processing. These are called *operators* at the bottom of Figure 1. CR equipments must be flexible, for that they should contain *operators* with the ability to be reconfigured or changed in one or more of their parameters during operation. This reconfiguration must comply with real-time constraints, and should not undermine the proper operation of the system Hence the need to include a sub-part for reconfiguration management named *ReM(u)* for reconfiguration managers (units) [2]. Moreover, cognitive radio equipments must interact with their environment, hence the need for sensing operators. These operators gather information and then send metrics to a cognitive sub-part able to make decisions following the arrival of these metrics. This part for cognitive management is made of *CRM(u)* for cognitive radio managers (units). *CRM(u)* analyze data and give adequate orders of reconfiguration to the (flexible) processing operators, through the reconfiguration managers.

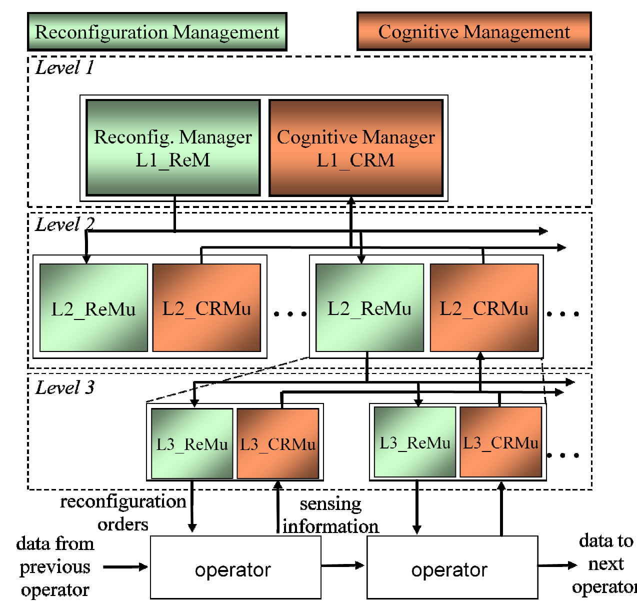

Figure 1 - HDCRAM Architecture deployment example

Three levels are depicted in Figure 1: one top level L1 acting as the head of the system, the lowest level made of L3 managers, one for each operator, and at least one intermediate level with several managers, the number of which depends on the processing functionality or repartition. That is why HDCRAM stands for Hierarchical and Distributed Cognitive Radio Architecture Management. More can be found in [15].

4.2. PIM Modeling Tools

4.2.1 From an architecture to a meta-model

After defining these rules, it is important to formalize them. A meta-model has been created in that purpose, as a template to generate the model of any CR equipment. It is essential that the models generated from this meta-model would not be affected by the technological leaps so that we only consider in this section a PIM meta-model of HDCRAM. One more thing is that the formalism is as understandable as possible by the radio design community, hence the choice of MOF as a formalism to define the HDCRAM meta-model. Indeed MOF is the most popular and known meta-model standard. It has been used to describe the meta-model of UML as well as several profiles such as SysML or MARTE for embedded systems [10]. Moreover, using MOF instead of using directly UML is a guarantee to have a restricted and rigorous CR model (compared to UML). It also gives the opportunity to take benefit from UML semantics in which we only borrow the necessary concepts selected in the meta-model.

4.2.2 HDCRAM meta-model

At this stage, we have implemented HDCRAM PIM meta-model in GMF, which is shown in Figure 2. The graphical implementation is possible either thanks to UML class diagram creation tool, as a tree via *ecore* model, or as text in XMI (XML Metadata Interchange). XMI is an OMG standard for formatting models content and ensure compatibility between modeling tools.

In order to explain the global philosophy of the design approach, PDM and PSM are addressed in the next two sections. But they still are at the project level and have not been implemented yet.

4.2.3. From a meta-model to a modeling environment

The goal is to create a modeling environment specifically for cognitive radio equipments design. This environment would allow designers creating their diagrams following the rules and formalism imposed by HDCRAM meta-model. To create this environment, we need tools dedicated to the implementation of a Domain Specific Language (DSL) compatible with MOF. Several commercial tools that help making DSL are available such as MetaEdit+, Microsoft DSL, Obeo Designer or Poseidon for DSLs. Our choice is an open source tool that is available as a plugin for eclipse: GMF (Graphical Modeling Framework), which provides a framework for developing a graphical editor [19]. The implementation of a graphical editor with GMF goes through a design flow based on the definition of several domain models, tooling, graphical and mapping model definition.

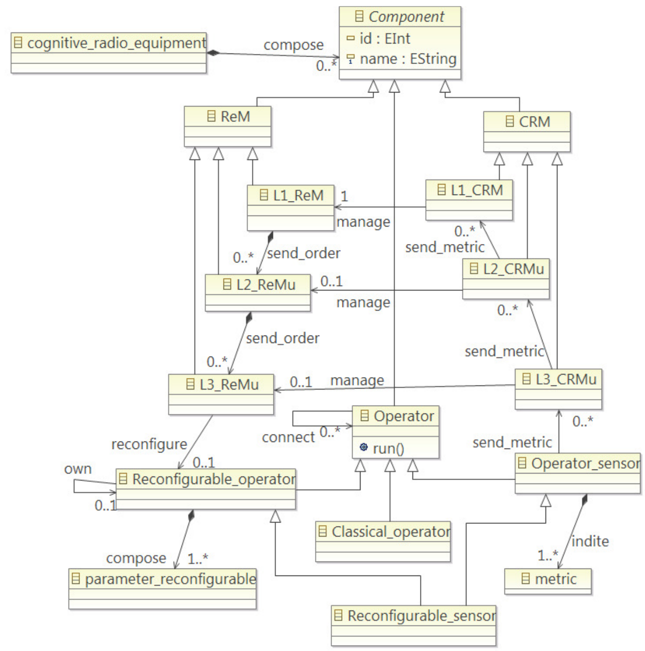

Figure 2 - HDCRAM Meta-Model

Figure 3 illustrates how to generate a fully functional graphical editor based on GMF. Each of the following paragraphs explains a particular step of this graph.

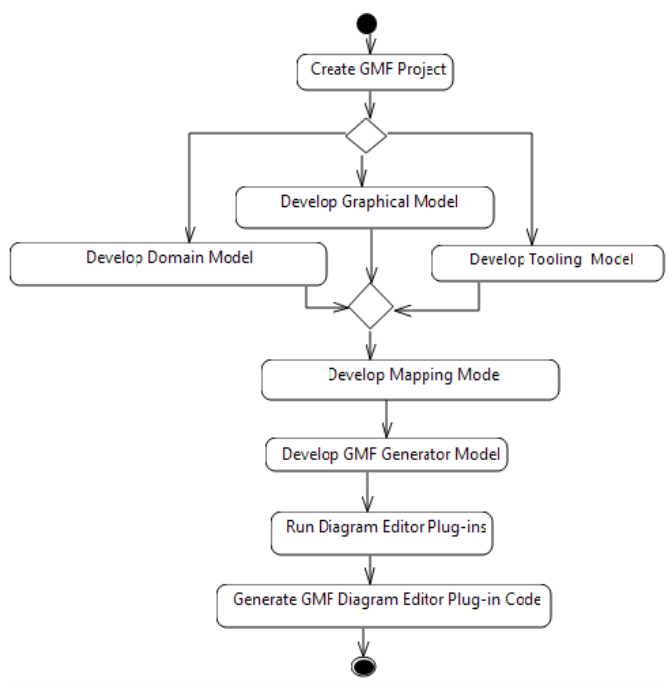

Figure 3 - GMF overview [19]

The Domain Model Definition is a conceptual model for describing the diagram generated by the editor. It describes the different entities included in the diagram, their attributes, roles and relationships. The HDCRAM meta-model provides the necessary specifications to describe any cognitive radio equipment.

The Graphical Definition determines the semantics of the generated diagrams. For our editor, we chose to reuse UML notations described by MOF for the presentation of node and path. Table 1 associates each component and relationship model in the graphical notation used by the editor. Table 1 shows the complete semantics currenty used in PIM models.

Table 1 – Definition of graphic node/path included in diagram

Node/Path	Notation	TYPE
Operator	ClassName	Class
ReM	ClassName	Class
CRM	ClassName	Class
manage		Association
connect		Dependency
send_metric		Association
reconfigure		Association
send order		Composition
implement		Generalization

The Tooling Definition allows customizing the editor by adding options in the bar menu for models validation. It also enables to add HDCRAM items into the palette, allowing a user to draw one's own model. The tooling definition therefore defines the tools which are going to be created by GMF in the editor.

The Mapping Definition consists in combining the three previous models into one. For this, we associate a meta-model element with a graphical form and finally with a component on the editor's palette. We apply a transformation to this model to obtain the final model. We then associate this final model with the Generating Model that creates Java source code plugin. We can also configure the runtime of the Eclipse plugin to launch a new workspace.

4.3. PDM Modeling Tools

Concerning CR systems, it is important to define the execution platform that runs the application itself. For the definition of platforms, we propose a diagram based on the UML deployment diagram. Indeed, the deployment diagram is often used to schematize the structure of organization of a computer system. It is very useful for distributed systems, since it allows presenting how the system components are distributed and how these components communicate.

To integrate this diagram in our modeler for CR equipments, we apply the same design flow as for PIM and we implement the platform meta-model of Figure 4. This meta-model is mainly composed of nodes and connections. Nodes correspond to a device or an execution environment. Devices are a physical system or a hardware component. Regarding the execution environment, it can be a software platform or an operating system. We also include the notion of artifact existing in the deployment diagram to present the elements outside the platform, which are essential to its execution, such as, configuration file for a program, or a bitstream of an FPGA. We also implement the different models necessary for the generation of the diagram using GMF:

- the domain model definition,
- the graphical definition,
- the tooling definition,
- the mapping definition needed to generate the diagram editor for modeling PDM platforms.

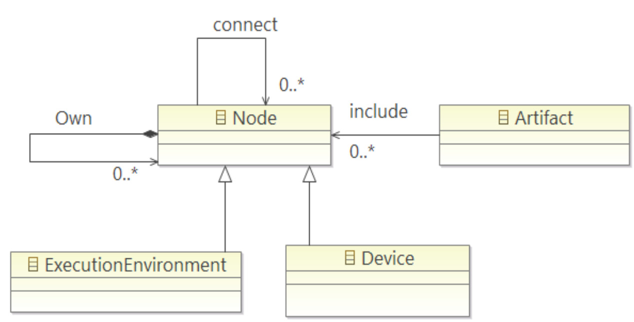

Figure 4 - Platform Meta-model

It is necessary to notice that unlike the PIM, the PDM is not unique: multiple PDM can be modeled for a single PIM [12].

4.4. PSM tools

The PSM is a formal description of the system; it is the result of integration between a PIM and a PDM. This obviously means that we can have as many PSM and PDM as desired, and that is the major advantage of the MDA approach. Indeed we are able to change the platform without changing anything in the application and even reuse an existing platform for new applications, in particular for maintenance. It is for sure easier to maintain models than maintaining lines of code.

Up to now, it remains to transform each component of the PSM to source code. For this purpose we chose to use ACCELEO, which is an open source tool compatible with our modeler for CR equipment based on GMF. ACCELEO [20] is based on templates to convert a model to text. The tool already offers templates for specific software platforms such as C or Java. It also offers the possibility of integrating its own templates to generate the C++, VHDL or System C. The interest of this model to text (M2T) transformation based on templates relies in the simplicity of the PDM models. Indeed, the development complexity is included in the templates, not in the models. It is imperative that the template may be reusable by the implementation of generic components. It is important to indicate that this approach could be coupled to others such as the model to model (M2M) transformation approach, which is more focused on models and based on modeling on any target platform (by including symmetric target language in the model).

5. EQUIPMENT MODELING FOR OSA SENARIO

The explosive growth of wireless services last recent years has created a serious congestion problem of radiofrequency spectrum [2]. This is the static allocation of spectrum that leads to the inefficient use of spectrum resources. Each band indeed is allocated to a primary user (PU), which prohibits other users to use it even in the absence of PU, creating that way white space (holes) in the spectrum [4]. The new US legislation has authorized the use of holes by a secondary user (SU) for a certain frequency band (TV broadcast) with restrictions [3]. In more futuristic scenari [2], we could imagine generalizing this approach to many other bands, with the constraint that the SU must leave the band as soon as the PU wants to use it again (which does not occur in TV band as transmission is permanent and static in frequency). The SU should then be able to interact with its environment, to detect the return of PUs, and then decide to release the band and to reconfigure its operation by changing frequency without interrupting the service. We show hereafter how can be used the proposed modeling environment for CR facilities to model that a SU should be able to accomplish the OSA scenario. In this example, SU is a video transmission system composed of a SU base station and SU terminal operating in a band in an unlicensed manner.

5.1. Base Station Model

The base station, subsequently named SU#1 (respectively terminal named SU#2), transmits (respectively receives) a video stream (via the User Datagram Protocol – UDP) with a simple BPSK mapping. In the demonstrator, baseband processing is executed in a computer, and radio frequency translation is done by a USRP platform from Ettus Research™ [21]. So each of base-station and terminal is composed of a computer and a USRP platform. Figure 5 shows the PIM model drawn for the base station without distinction of implementation target (USRP or computer). As the SU should be able to change of carrier frequency in this scenario, a reconfigurable parameter exists for the *CarrierTX* operator of SU#1 (respectively *CarrierRX* operator of SU#2). In our scenario, a radio reverse link operating at a licensed (so not concerned by OSA, without loss of generality) frequency enables to coordinate frequency jumps between SU#1 and SU#2 (SU#2 decides indeed [22]).

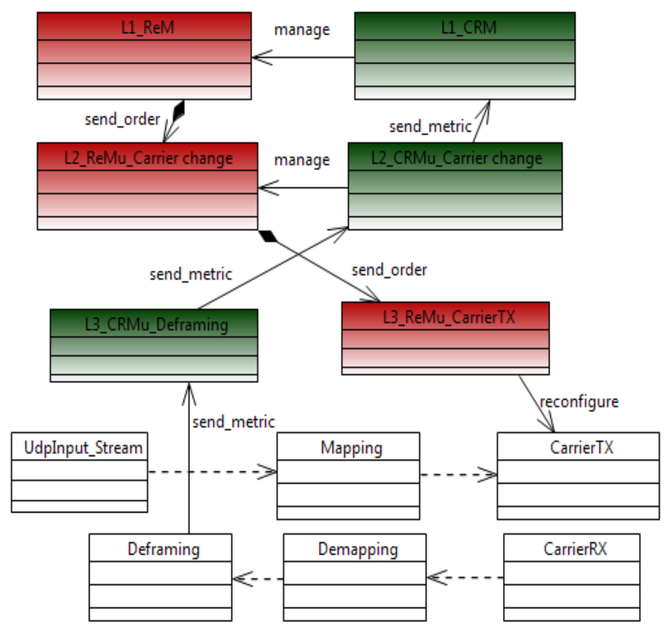

Figure 5 - Base Station Model (SU#1)

The *Deframing* operator of SU#1 recovers the instruction sent by the terminal and sends it to HDCRAM management. *Deframing* operator acts as a sensor that is responsible for transferring the received instruction from SU#2 ("frequency jump" order in the current scenario) to the cognitive unit *L3_CRMu_deframing*. The *L3_CRMu_deframing* interprets the metric indicating (or not) the return of a PU at the frequency used by SU. If it is positive (a PU comes back at the SU frequency), it then informs its higher cognitive unit *L2_CRMu_Carrier change*. At this level the decision to change the frequency transmission is taken and the reconfiguration order is sent to the reconfiguration manager *L2_ReMu_Carrier change*. For this scenario only operator *CarrierTX* is reconfigured in SU#1. We see in Figure 5 that only *CarrierTX* and *Deframing* operators are connected to HDCRAM management as they are the only operators to play a role of sensor or renconfigurable operator. In more complex scenari, *L2_ReMu* would have a list of all the operators to reconfigure and would give orders of reconfiguration to respective *L3_ReMu* in a way to avoid the service rupture.

5.2. Terminal

The SU terminal SU#2 receives the video stream sent by the base station SU#1. SU#2 performs a demapping operation before sending back the data to the Video LAN Client (VLC) which plays a streaming video via a UDP socket. The video sent by SU#1 is displayed on the terminal, and to detect the return of PU, a sensor is added in the model. We implement here a simple energy detector as a sensor which sends metrics to its hierarchical cognitive *L3_CRMu_Detector* that informs the *L2_CRMu_Carrier change* responsible for the management of OSA scenario. *L2_CRMu_Carrier change* knows that when it is informed that the PU comes back at the current frequency, it immediately not only changes its carrier frequency, but also informs SU#1 to change its frequency too, so that the SU link jumps at a free frequency (pre-fixed if reverse link is licensed or could be derived using learning techniques of [14] if reverse link also works in OSA mode). To change its own frequency, an order of reconfiguration is sent to *L3_ReMu_carrierRX*. In order to inform SU#1 about the frequency jump, another order of reconfiguration is sent to *L3_ReMu_Framing*, so that this information is incorporated in the next frame sent to SU#1 by the *Framing* operator. Then we see in Figure 6 that three operators have management entities: *Detector*, *CarrierRX* and *Framing*.

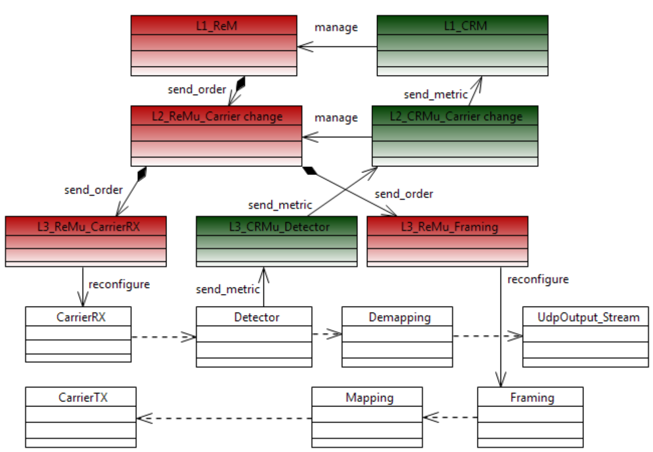

Figure 6 - Terminal Model (SU#2)

5.3. Demonstration Platform model

Figure 7 shows a model of the platform used for the implementation of the OSA scenario exposed above. For this scenario, the system needs to run on machines equipped with an x86 windows 7 environment. Moreover, we also used the diagram to show the different framework, QT framework to permit multithreading, UHD is the Windows driver of USRP platforms. On each side (SU#1 and SU#2) the computer is a host for USRP platform through a Gigabit Ethernet link.

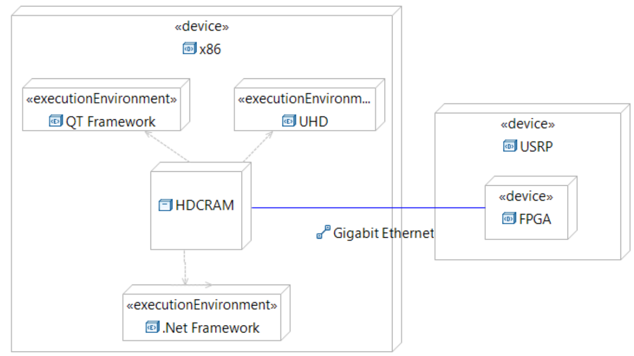

Figure 7 - Deployment diagram for OSA demonstration

This scenario has been manually implemented and the demonstration runs as planned. The new tooling proposed in this paper should soon enable to automatically generate it so that we can compare both design approaches. Note that automatically generating the code for this demonstration is is quite straightforward as the entire execution in this context is done in the host computer, so only homogeneous code is required (typically C here) to manage both baseband processing on the laptop and the USRP platform through UHD API. The longer term will consist in generalizing the design approach to heterogeneous platforms.

6. CONCLUSION

This paper presents an approach for designing cognitive radio equipment with a high level of abstraction. This approach aims in the long term at generating code for heterogeneous platforms through a modeling process gathering application modeling and platform modeling in a MDA approach. We insist here on the modeling aspects of the approach, not the code generation, which will be the scope of a future paper. We also present the tools that were developed or used for each design stage. The scenario described has been implemented and runs as a demonstrator with hand-made code. The comparison with automatic code generation provided by the development environment will enable to validate and evaluate its performance in real conditions

7. ACKNOWLEDGMENT

Authors would like to thank the Brittany Region (Région Bretagne) for their support in this work.

8. REFERENCES

[1] J. Mitola, “Cognitive Radio” ” Licentiate proposal, KTH, Stockholm, Sweden, Dec. 1998.

[2] J. Palicot, "Radio Enineering: From Software radio to Cognitive Radio", Wiley 2011; ISBN: 978-1-84821-296-1

[3] D. Noguet, M. Gautier, V. Berg, " Advances in opportunistic radio technologies for TVWS "EURASIP Journal on Wireless Communications and Networking 2011

[4] Q. Zhao, A. Swami, "A Survey of Dynamic Spectrum Access: Signal Processing and Networking Perspectives", in IEEE ICASSP: special session on Signal Processing and Networking for Dynamic Spectrum Access, April, 2007

[5] OMG, Object Management Group, "MDA Guide Version 1.0.1", http://www.omg.org/docs/omg/03-06-01.pdf, June 2003

[6] OMG, "PIM and PSM for Software Radio Components", http://www.omg.org/docs/dtc/04-05-04.pdf, Final Adopted Specification

[7] M. Barbeau, F. Bordeleau, J. Smith, "An Introduction to a UML Platform Independent Model of a Software Radio", International Conference on Telecommunications (ICT), Beijing, 2002

[8] S. M. Bicer, "A Software Communications Architecture Compliant Software Defined Radio Implementation", PhD Thesis, Northern University, Boston, USA, June 2002

[9] S. Rouxel, J.P. Diguet, N. Bulteau, J. Carre-Gourdin, J.E. Goubard, C. Moy, "UML framework for PIM and PSM verification of SDR systems", SDR Forum Technical Conference’05, Anaheim, USA, November 2005

[10] M. Colizza, C. Rinaldi, I.J. Senese, "A generalized waveform identifier technique for Software Radio", Communications Control and Signal Processing (ISCCSP), Rome, Italy, 2-4 May 2012

[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 Journal, Vol. 53, Issues 3-4, pp. 471-484, Feb. 2011

[12] OMG, "MOF Model to Text Transformation Language Request for Proposal", http://www.omg.org/docs/ad/04-04-07.pdf, May 2004

[13] OMG, "Meta Object Facility (MOF) 2.0 Core Specification", http://www.omg.org/docs/ptc/04-10-15.pdf, October 2004

[14] W. Jouini, C. Moy, J. Palicot, "Decision making for cognitive radio equipment: analysis of the first 10 years of exploration", *EURASIP Journal on Wireless Communications and Networking 2012*, 2012:26.

[15] L. Godard, C. Moy, J. Palicot, "An Executable Meta-Model of a Hierarchical and Distributed Architecture Management for the Design of Cognitive Radio Equipments", Annals of Telecommunications special issue on Cognitive Radio, vol. 64, pp.463-482, n°7-8, Aug. 2009

[16] JTRS, since October 2012: JTNC, http://jtnc.mil/sca

[17] M. Usman, A. Nadeem, T. Kim, E. Cho, "A Survey of Consistency Checking Techniques for UML Models", in Advanced Software Engineering and Its Applications (Los Alamitos, 2008), IEEE Computer Society, 57-62

[18] W. J. Dzidek, L. C. Briand, Y. Labiche, "Lessons Learned from Developing a Dynamic OCL Constraint Enforcement Tool for Java", In: Proc. MODELS 2005 Workshops, LNCS, 3844 (2005) 10-19

[19] Eclipse GMF - Graphical Modeling Framework, Official Web Site http://www.eclipse.org/modeling/gmp

[20] J.Musset, E. Juliot, S Lacrampe,”Acceleo User Guide.”http://acceleo.org/doc/obeo/en/acceleo-2.6-user-guide.pdf, 2.6 edition, 2008

[21] Ettus Research “Products” - accessed 02/04/2012 http://www.ettus.com/products

[22] O. Lazrak, P. Leray, C. Moy, "HDCRAM Proof-of-Concept for Opportunistic Spectrum Access", DSD Conference, Cesme, Turquey, 5-8 September 2012