

Méthodologie et outils de conception des équipements de radio intelligente

OUSSAMA LAZRAK, CHRISTOPHE MOY, PIERRE LERAY

IETR/SUPELEC

Campus de Rennes, Avenue de la Boulais, CS 47601, F-35576 Cesson-Sévigné Cedex, France
{oussama.lazrak, christophe.moy, pierre.leray}@supelec.fr

Résumé – Dans cet article nous proposons une approche d'ingénierie dirigée par les modèles pour la conception des équipements de radio intelligente. L'approche consiste à décrire le système à différents niveaux d'abstraction allant d'une représentation à haut niveau d'abstraction jusqu'au code qui sera exécuté sur une plateforme spécifique. Cette plateforme peut être une plateforme matérielle à base de FPGAs et ASICs, et/ou logicielle de type Corba, .Net ou Java. Cette approche nécessite l'usage d'un langage de modélisation propre à la radio intelligente, mais qui est quand même compréhensible par la plupart des concepteurs d'équipements de radio intelligente. Associé à ce langage, nous proposons un environnement de conception fournissant les outils nécessaires à la réalisation des modèles. Ce travail se base sur nos travaux antérieurs, notamment sur l'architecture HDCRAM.

Abstract - A Model Driven Engineering (MDE) approach is proposed in this paper for the design of Cognitive Radio (CR) equipments. Model Driven Architecture (MDA) approach consists in describing a system at different levels of abstraction, from a high level abstraction representation (often graphical) down to 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

L'utilisation d'une approche d'ingénierie dirigée par les modèles (IDM) consiste à appliquer plusieurs transformations à un modèle pour obtenir du code exécutable relatif à une plateforme cible. Cette approche est parfaitement adaptée à la conception des systèmes embarqués parce qu'elle permet de décrire séparément les deux parties constituantes d'un système électronique [2]: l'application et la plate-forme matérielle d'exécution. L'application peut être définie à l'aide de standards comme UML ou Merise. La plateforme matérielle peut être plutôt définie par des techniques plus appropriées telles que MARTE ou SysML. Ces deux modèles nommés PIM (*Platform Independent Model*) et PM (*Platform Model*) sont par la suite assemblés dans un seul modèle dépendant de la plateforme PSM (*Platform Specific Model*). Il s'agit donc de construire des modèles (PIM) de systèmes, de façon à prendre en compte tous les besoins fonctionnels (sans introduire des notions relatives à la plate-forme, aux langages de programmation ou autres aspects liés à la réalisation), puis de cibler un environnement d'exécution matériel spécifique (DSP, FPGA, GPP...) ou logiciel (Corba, .net, Java...). L'approche IDM permet également de traduire un modèle indépendant de toute plateforme (PIM) en un modèle spécifique à une plateforme (PSM) à l'aide d'outils automatisés, facilitant la migration vers de nouvelles technologies, et permettant un gain considérable de temps pour mettre en œuvre un système. Ces outils sont principalement des outils de transformation de modèles parmi lesquels nous

pouvons citer : ATL (*Atlas Transformation Language*), BOTL (*Basic Object-oriented Transformation Language*) ou encore Aceleo. Dans cet article, nous nous intéressons essentiellement à la modélisation PIM en présentant notre approche de conception des équipements de radio intelligente [3] basés sur notre méta-modèle HDCRAM (*Hierarchical and Distributed Cognitive Radio Architecture Management* [4]), ainsi qu'à la présentation d'un environnement de modélisation dédié aux équipements et composants de radio intelligente.

2 Modélisation des équipements de radio intelligente

Pour concevoir des équipements radio, on a besoin d'une architecture décrivant le système, d'un langage compréhensible par les différents intervenants lors de la phase de mise en œuvre, ainsi que d'un environnement de conception offrant les outils nécessaires à la réalisation du modèle. Ce dernier doit pouvoir prendre en considération les besoins spécifiques au concepteur d'une radio intelligente. Ces besoins ont été décrits dans le méta modèle HDCRAM qui peut être vu comme une matrice capable de générer des modèles respectant la forme imposée par le méta modèle. Il est important également que HDCRAM soit formalisé et décrit, d'où le choix de MOF (Meta Object Facility). En effet, MOF [5] est le standard de méta modélisation le plus utilisé et le plus connu. Il a été utilisé pour décrire le méta modèle d'UML ainsi que plusieurs profils comme SysML ou MARTE réalisé par l'OMG, ou même un

profil UML dédié à la radio logicielle [6][7]. HDCRAM représente l'élément essentiel pour le développement d'un environnement de modélisation des équipements et des composants de radio intelligente. Cet environnement disponible sous forme d'un *Plugin* pour l'environnement Eclipse est essentiellement composé d'un éditeur de diagrammes, d'un outil de validation et d'identification des erreurs de modélisation et d'une palette contenant les différentes entités et relations possibles dans un diagramme basé sur HDCRAM.

2.1 L'architecture HDCRAM

L'architecture HDCRAM décrit d'une manière symbolique et schématique les différents éléments de l'architecture de gestion existants dans un équipement de radio intelligente, leurs interrelations et leurs interactions. Contrairement aux spécifications, le modèle d'architecture ne décrit pas ce que doit réaliser le système mais plutôt quelle structure de gestion il doit contenir en plus des opérateurs de la chaîne de traitement radio pour répondre aux spécifications en termes de scénarios de radio intelligente qu'il doit supporter. La spécification est basée sur le cycle intelligent proposé par J. Mitola [8] qui affirme que les équipements radio du futur devraient être plus autonomes grâce à l'utilisation de technologies flexibles, ce qui leur permettra de modifier dynamiquement leurs fonctionnalités. Les équipements radio doivent également être capables de capter des informations sur leur environnement, de détecter certaines variations, et de prendre aussitôt la bonne décision d'adaptation à ces changements. Ceci amène à proposer une architecture induisant à la fois des unités responsables de la reconfiguration et d'autres pour la prise de décision.

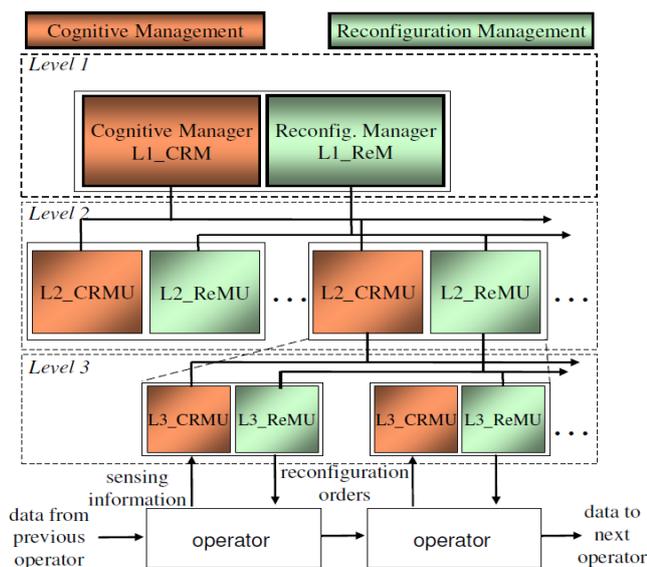


Figure 1: Aperçu d'un déploiement HDCRAM

Ces éléments interagissent sur les opérateurs de traitement de signal afin de permettre à l'équipement de réaliser des scénarios de plus en plus complexes, de manière autonome. La reconfiguration et la prise de

décision se basent souvent sur des algorithmes et des techniques qui peuvent se montrer assez complexes, on a choisi de les répartir sur trois niveaux, chacune de ces unités étant responsable de la gestion et du bon fonctionnement des unités inférieures. Cette répartition s'avère également très efficace pour les équipements hétérogènes incluant à la fois des FPGAs, GPPs. La Figure 1 illustre un déploiement de l'architecture HDCRAM représenté au-dessus de 2 opérateurs d'une chaîne radio.

2.2 Environnement de modélisation d'équipements de radio intelligente

Nous souhaitons réaliser un environnement de modélisation spécifique à la radio intelligente qui permettra aux concepteurs de systèmes de créer leurs modèles tout en respectant les règles et le formalisme imposés par le méta-modèle HDCRAM. Pour créer cet environnement, des outils permettant la mise en œuvre de DSL (*Domain Specific Language*) compatibles avec MOF doivent être utilisés. Plusieurs outils commerciaux permettant de réaliser des DSL existent sur le marché, comme MetaEdit+, Microsoft DSL tools ou Obeo Designer. Notre choix s'est porté sur un outil open source intégré à l'environnement Eclipse, GMF [9] (*Graphical Modeling Framework*) qui fournit un support pour le développement des éditeurs graphiques de modélisation. La réalisation d'un éditeur passe par un flux de conception se basant sur plusieurs modèles. Chacun de ces modèles décrit un aspect de l'éditeur.

2.2.1 Domain Model Definition

C'est le modèle conceptuel décrivant les diagrammes supportés par l'environnement. Il décrit les différentes entités, leurs attributs, les rôles et les relations, ainsi que les contraintes imposées. Il suffit donc de mettre en œuvre le méta-modèle HDCRAM sous Eclipse et de générer le fichier *ecore* correspondant. Eclipse fournit différentes façons de visualiser le méta-modèle, soit sous forme graphique grâce à l'outil d'affichage de diagramme UML, ou bien sous forme textuelle en XMI (*XML Metadata Interchange*).

2.2.2 Graphical definition

Le *Domain Model* définit les entités et les relations existantes dans l'éditeur mais il ne définit pas comment elles seront représentées graphiquement. La *graphical definition* permet donc de définir la sémantique des diagrammes générés. Pour notre éditeur, on a choisi de réutiliser les notations graphiques d'UML. La Table 1 associe à chaque composant et relation dans le domaine model la notation graphique utilisée par l'éditeur.

2.2.3 Tooling definition

Ce modèle permet de personnaliser l'éditeur en ajoutant les options de validation de modèle au menu et il permet de créer la palette et d'y inclure les composants HDCRAM. Ceci permet à l'utilisateur d'ajouter les éléments nécessaires à son scénario dans le

diagramme et de valider le diagramme résultant. Il est à noter que l'éditeur refuse toute relation ou entité non existante dans le méta modèle, ce qui garantit que les modèles sont bien fidèles aux règles imposées par le méta modèle.

Table 1: notation graphique propose par le modeleur

Entité/Relation	Notation	Type
Operator		Classe
ReM		Classe
CRM		Classe
Manage		Association
Connect		Dependence
send_metric		Association
reconfigure		Association
send order		Composition
implement		Généralisation

2.2.4 Mapping Definition et génération de l'environnement

Cela consiste à combiner les trois précédents modèles en faisant correspondre chaque élément de HDCRAM à une forme graphique puis à un élément dans la palette. Il suffit ensuite de générer le modèle final, le code source en java ainsi que le *plugin*.

3 Exemple de Modélisation d'équipements de radio intelligente

La croissance rapide des services sans fil ces dernières années a créé un problème d'encombrement du spectre radiofréquence. De plus, l'allocation statique du spectre conduit à une utilisation inefficace de celui-ci. En effet chaque bande est affectée à un utilisateur principal (PU), qui interdit aux autres utilisateurs d'utiliser sa fréquence même en son absence, laissant ainsi des espaces libres (trous) mais inutilisés dans le spectre. La nouvelle législation américaine a autorisé l'utilisation de ces trous par un utilisateur secondaire (SU) pour une certaine bande de fréquence (émission TV) avec des restrictions [10]. Pour des scénarios futuristes, nous pourrions imaginer que des équipements radio soient en mesure d'exploiter les trous laissés par les PU avec la contrainte que tout SU doit quitter la bande dès que le PU veut l'utiliser à nouveau [11]. Le SU devrait alors être en mesure d'interagir avec son environnement, de détecter le retour du PU, et ensuite de décider de libérer la bande et de s'auto-reconfigurer pour changer sa fréquence sans interrompre le service. Nous montrons ci-dessous comment on peut utiliser l'environnement de modélisation proposé pour modéliser un SU capable d'accomplir le scénario OSA. Dans cet exemple, le SU est un système échangeant un flot vidéo composé d'une station de base et d'un

terminal. Le SU fonctionnant dans une bande d'une manière non licenciée.

3.1 Station de base

La station de base transmet un flux vidéo (via le *User Datagram Protocol - UDP*) avec une constellation BPSK simple. Dans la démonstration, le traitement en bande de base est exécuté sur un GPP d'ordinateur, et la traduction des fréquences radio se fait par une plateforme USRP de Ettus Research™ [12]. Ainsi, la station de base et le terminal sont composés chacun d'un ordinateur et d'une plate-forme USRP. La Figure 2 montre le modèle réalisé pour la station de base sans distinction de la cible mise en œuvre (USRP ou ordinateur). Comme le SU doit être capable de changer de fréquence porteuse dans ce scénario, il existe un paramètre reconfigurable pour l'opérateur *CarrierTX*, de même pour l'opérateur *CarrierRX* du Terminal.

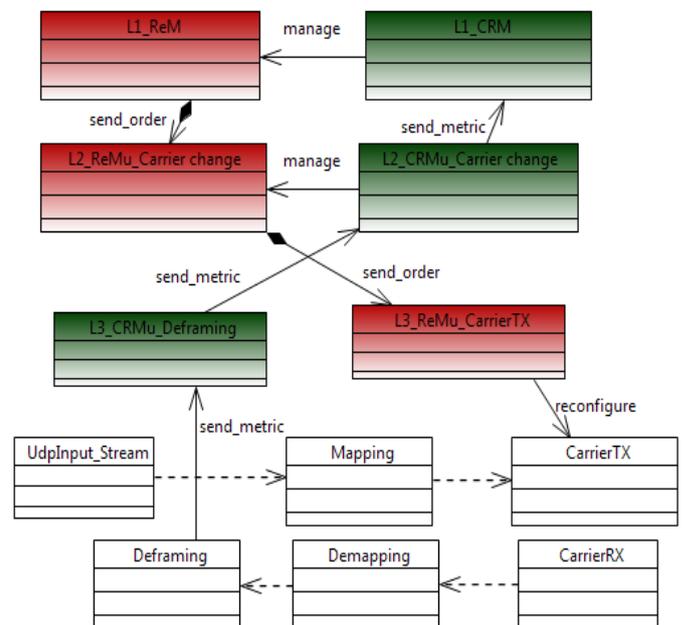


Figure 2: Modèle de la Station de base OSA

3.2 Terminal

Le terminal reçoit le flux de données vidéo envoyées par la station de base. Comme illustré dans la Figure 3, il réalise une opération de *demapping* avant de renvoyer les données à un client de lecture vidéo VLC par l'intermédiaire d'une connexion UDP. La vidéo envoyée par la station de base est donc affichée sur le terminal, et pour détecter le retour du PU, un capteur est ajouté dans la chaîne de réception radio. Nous mettons en place ici un détecteur d'énergie. Ce capteur envoie des mesures à son gestionnaire *L3_CRMu_Detector* de niveau trois, qui informe à son tour le gestionnaire de niveau deux *L2_CRMu_Carrier* responsable de la gestion de scénario OSA. *L2_CRMu_Carrier* a été conçu pour que lorsqu'il est informé du retour de PU, il doit immédiatement modifier sa fréquence de réception, mais aussi informer la station de base pour qu'elle modifie sa fréquence d'émission aussi, de telle sorte que la connexion entre les deux équipements ne soit pas interrompue. Pour changer sa propre fréquence, un

ordre de reconfiguration est envoyé au gestionnaire de reconfiguration *L3_ReMu_carrierRX*. Afin d'informer la station de base sur le saut de fréquence, un autre ordre de reconfiguration est envoyé au gestionnaire de reconfiguration *L3_ReMu_Framing*, de sorte que la mise en trame soit reconfigurée pour incorporer cette information de contrôle dans la prochaine trame envoyée à la station de base.

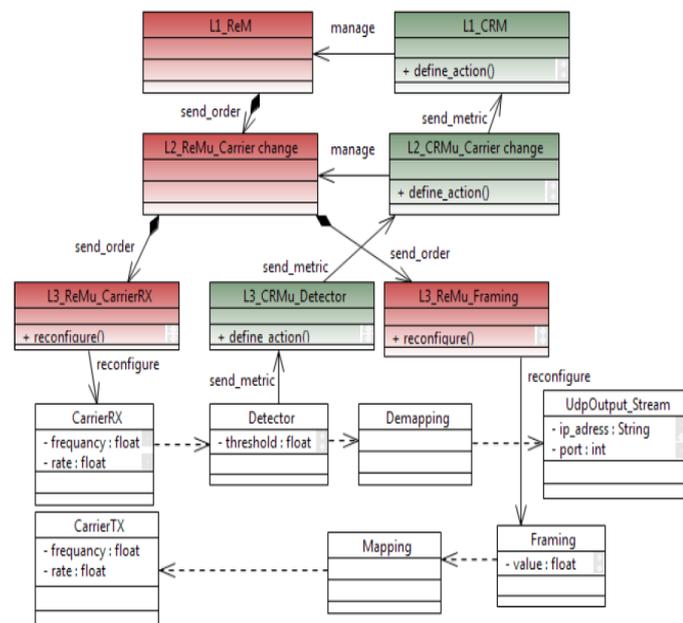


Figure 3: Modèle du terminal OSA

4 Génération automatique de code

L'un des avantages de proposer un modèleur sous forme de plugin Eclipse est le fait de pouvoir utiliser d'autres plugins existants. Eclipse en propose plusieurs, et leurs utilités sont diverses, comme Kermeta qui est un plugin permettant la simulation des modèles, ou OxygenXML qui est un outil de génération de la documentation. Cependant, nous nous sommes intéressés à un outil permettant de générer du texte à partir d'un modèle. Cet outil, nommé Acceleo [13] est un outil *open source* compatible IDM, qui est capable d'intégrer tout PIM écrit en XMI. Il prend donc en charge les modèles générés par notre modèleur. Pour le modèle de la plateforme PM, Acceleo fournit déjà plusieurs Template pour supporter des langages tels que JAVA, C ou C#. Il offre également la possibilité d'intégrer d'autres plateformes par la création de nouveaux Template pour générer du code C++, VHDL ou SystemC... Nous avons choisi de créer nos propres *Templates* pour générer l'architecture HDCRAM telle qu'elle était définie lors de la spécification. En effet, on peut remarquer que le modèle est une représentation architecturale de l'équipement (un diagramme de classe UML), il représente le système d'un point de vue statique en mettant l'accent sur ce qui devrait exister dans le système. Le fonctionnement interne de chacun des blocs n'est pas détaillé dans ce diagramme. La génération n'inclut donc que l'architecture, quant aux fonctionnements, ils seront réalisés via des bibliothèques

externes ou bien générés à partir d'autres diagrammes, comme un diagramme de séquence ou bien un diagramme d'état-transitions.

5 Conclusion

Les équipements radio deviennent de plus en plus intelligents et intègrent de plus en plus de fonctionnalités, mais ils sont par conséquent plus difficiles à concevoir. Sachant que la phase de conception des systèmes complexes est la phase la plus coûteuse en temps et en argent, des méthodologies et des outils spécifiques sont devenus nécessaires. Comme la radio intelligente utilise la radio logicielle comme support, il est donc évident qu'elle peut aussi bénéficier de certaines techniques utilisées jusqu'à présent dans le monde informatique, notamment la modélisation à haut niveau d'abstraction qui vise la génération automatique de code. Cet article propose un outil concret destiné à la modélisation des équipements radio intelligents, intégré dans un environnement de développement couramment utilisé. Nos méthodes et outils ont été testés pour réaliser des démonstrations, notamment un équipement supportant un scénario d'accès opportuniste au spectre.

6 Références

- [1] OMG, Object Management Group, "MDA Guide Version 1.0.1", <http://www.omg.org/docs/omg/03-06-01.pdf>, June 2003
- [2] OMG, "PIM and PSM for Software Radio Components", <http://www.omg.org/docs/dtc/04-05-04.pdf>, Final Adopted Specification
- [3] J. Palicot, "Radio Engineering: From Software radio to Cognitive Radio", Wiley 2011; ISBN: 978-1-84821-296-1
- [4] 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
- [5] OMG, "Meta Object Facility (MOF) 2.0 Core Specification", <http://www.omg.org/docs/ptc/04-10-15.pdf>, October 2004
- [6] 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
- [7] S. Rouxel, J.P. Diguët, 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
- [8] J. Mitola, "Cognitive Radio" Licentiate proposal, KTH, Stockholm, Sweden, Dec. 1998.
- [9] Eclipse GMF - Graphical Modeling Framework, Official Web Site <http://www.eclipse.org/modeling/gmf>
- [10] D. Noguet, M. Gautier, V. Berg, "Advances in opportunistic radio technologies for TVWS" *EURASIP Journal on Wireless Communications and Networking* 2011
- [11] 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
- [12] Ettus Research "Products" - accessed 02/04/2012 <http://www.ettus.com/products>
- [13] 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