

## **MODA : une architecture multimédia dirigée par les ontologies pour des systèmes multimédia en réseau**

Myriam Lamolle\*, Jorge Gomez\*\*, \*\*\*  
Ernesto Exposito\*\*

\*LIASD, IUT de Montreuil, Université Paris8,  
140 rue de la Nouvelle France, 93100 Montreuil  
m.lamolle@iut.univ-paris8.fr

<http://www.iut.univ-paris8.fr/recherche/LINC>

\*\*LAAS-CNRS, 7 Ave du Colonel Roche, 31077 Toulouse  
Université de Toulouse, UPS, INSA, INP ISAE, LAAS  
{jorge.gomez, ernesto.exposito}@laas.fr

\*\*\*Facultad de Matemáticas, Universidad Autónoma de Yucatán

**Résumé.** Les environnements multimédia actuels étant très hétérogènes, l'interopérabilité des systèmes multimédia mis en réseau et le déploiement automatique de tels réseaux multimédia sont très difficiles. En effet, la diversité des langages, des protocoles et des plateformes génèrent des problèmes d'incompatibilité très importants. De plus, l'instanciation et la configuration des systèmes multimédia guidées par les préférences et/ou les exigences des utilisateurs de façon dynamique n'est guère possible dans ce contexte. Cette étude introduit un cadre de travail appelé MODA fondé sur des ontologies pour le déploiement et la configuration automatiques de systèmes multimédia en réseau. S'appuyant sur des standards bien connus, MODA intègre les concepts du multimédia, des systèmes distribués et des plateformes logicielles. Nous présentons une étude de cas montrant comment un système multimédia en réseau peut être déployé dynamiquement afin d'illustrer les avantages inhérents à MODA.

## **1 Introduction**

Le développement exponentiel d'Internet et la diversité croissante des composants utilisables en réseau disponible sur le marché a entraîné un développement très rapide d'applications multimédia telles que IPTV, VoIP, VoD, visio-conférence, etc. Ces applications sont, pour l'heure, déployées de façon statique. De plus, les utilisateurs de ces applications appartiennent à des groupes ou des communautés bien définis (e.g., des utilisateurs d'un fournisseur internet recevant IPTV ou d'un groupe de travail qui communiquent au travers d'une visio-conférence). Avec l'explosion des réseaux sociaux, de nouvelles exigences sont apparus récemment pour intégrer des architectures logicielles ou physiques ainsi que de nouvelles

MODA : une architecture dirigée par les ontologies

interfaces permettant l'échange en direct d'informations multimédia entre des communautés ad hoc et des communautés mobiles. Ces nouvelles exigences soulèvent deux problèmes majeurs : l'interopérabilité et la dynamique. Premièrement, les systèmes multimédia actuels sont fortement hétérogènes. En effet, les applications multimédia sont implémentées dans des langages de programmation très variés (i.e., Java, C/C++, .net, etc.), des environnements logiciels différents (i.e., J2ME, Android, iPhone, JMF) et une pléthore de protocoles sont utilisés pour contrôler, représenter et échanger des données multimédia. D'autre part, les caractéristiques concernant les réseaux telles que la sécurité, les pare-feux, NAT, la mobilité et la qualité de service ont un impact important lors de sessions multimédia limitées temporellement. Elles doivent être prises en compte afin de garantir la satisfaction des préférences et/ou des exigences des utilisateurs finaux. Deuxièmement, la configuration et le déploiement de systèmes multimédia distribués devraient être dirigés par les exigences des utilisateurs dans le but de faciliter l'établissement spontané de sessions multimédia permettant l'échange d'informations entre des membres d'un réseau social par exemple.

Plusieurs recherches ont été menées afin de proposer des cadres de travail et des plateformes standardisés pour concevoir et développer des systèmes multimédia. Des organisations telles que l'ITU, l'IETF ou le W3C ont mis en place des groupes de travail pour mener à bien ces standardisations. D'autre part, les nombreuses plateformes de développement pour des composants intégrés ou mobiles sont constamment mises à jour.

Cependant, malgré ces efforts conséquents de standardisation, un cadre générique multimédia intégrant les concepts de base des systèmes multimédia distribués n'est toujours pas défini. Un tel cadre générique devrait être capable de représenter les différentes configurations des systèmes multimédia distribués. Il devrait aussi aider à configurer et déployer automatiquement de nouveaux systèmes selon les exigences et les préférences d'un utilisateur ou d'une communauté d'utilisateurs.

Notre travail de recherche consiste donc en la spécification et le développement d'une architecture multimédia dirigée par les ontologies (dite MODA). MODA est fondée sur les standards bien connus relatifs aux concepts spécifiques du multimédia, de la distribution et des plateformes. MODA inclut aussi un processus automatique s'appuyant sur l'utilisation de ces ontologies afin de rendre dynamique la traduction des requêtes d'un utilisateur ou d'une communauté d'utilisateurs pour le déploiement des composants multimédia. Les spécifications produites par notre architecture concernant le déploiement peuvent aussi être utilisées pour une auto-configuration du système de communication en déployant les composants de communication adéquats au niveau intergiciel, transport et réseau dans le but de fournir la qualité de service exigée pour garantir les exigences d'une session multimédia.

Cet article introduit dans la section 2 les différents cadres de travail orientés technologies et multimédia fournissant les fondements de MODA. La section 3 présente MODA, les ontologies de base et le processus de spécification des multimédia à déployer. La section 4 illustre par une étude de cas comment un système multimédia en réseau peut être déployé dynamiquement. Enfin, la section 5 présente nos conclusions et les perspectives de ce travail.

## 2 État de l'art

MODA est une architecture qui a été conçue en prenant en compte deux aspects à savoir, d'une part, la notion d'ingénierie dirigé par les modèles puisque divers standards co-existent dans le monde des réseaux multimédia et vont donc donner lieu à différents modèles ; d'autre part, la représentation sous forme d'ontologies des différents standards, protocoles et implémentations dans le domaine des systèmes multimédia pour faciliter l'interopérabilité.

### 2.1 Ingénierie dirigé par les modèles

L'objectif principal d'une approche d'ingénierie dirigée par le modèles (IDM) (Bezivin, 2004) est de reporter la complexité d'implémentation d'une application au niveau de sa spécification. Cela devient alors un problème d'abstraction de langage de programmation en utilisant un processus de modélisation abstrait fondé sur l'utilisation de plusieurs standards tels que MOF, OCL, UML et XMI.

L'architecture dirigé par les modèles (MDA) (Miller et Mukerji, 2003) (Poole, 2001) est un champ spécifique de l'IDM pour spécifier une architecture à 3 niveaux :

- une représentation d'un modèle indépendant (CIM) à partir d'un modèle. CIM décrit le contexte dans lequel les systèmes seront utilisés ;
- une représentation d'un modèle indépendant de toute plateforme (PIM) à partir de la représentation CIM. Elle décrit le système lui-même sans aucun détails sur l'utilisation de la plateforme. Une représentation PIM sera traduite pour une ou plusieurs plateformes architecturales réelles ;
- une représentation d'un modèle spécifique à une plateforme (PSM) à partir de la représentation PIM. A ce niveau, l'environnement des plateformes ou des langages d'implémentations sont connus.

MDA permet de concevoir un workflow à partir des différentes traductions depuis la représentation CIM jusqu'à la représentation PSM. Ces transformations peuvent être automatisées, notamment quand un métamodèle spécifie les règles de transformation entre les différents modèles impliqués. Par ce biais, une approche MDA augmente l'interopérabilité dans des environnements hétérogènes et fournit une méthode d'intégration de systèmes en utilisant des moteurs génériques de transformation.

Ces dernières années, le groupe de travail du W3C SWBPD<sup>1</sup> a proposé d'étendre la méthodologie MDA en utilisant les modèles sémantiques ou les ontologies (i.e., les langages RDF et OWL). Cette extension a été définie par une architecture dirigée par les ontologies (ODA) (voir ODA, 2001). ODA étend MDA en fournissant une représentation des vocabulaires du domaine non ambigus (e.g., exigences, contraintes, services, propriétés, etc.), le contrôle de la consistance du modèle et la validation aussi bien que la possibilité de choisir de nouvelles caractéristiques logicielles automatiquement. Plusieurs travaux ont été mis en exergue pour illustrer comment une approche basée-ontologie peut être utilisée dans la conception, le développement et la gestion de systèmes distribués. Par exemple, dans (Oberle *et*

---

<sup>1</sup> « Semantic Web Best Practices and Deployment »

MODA : une architecture dirigée par les ontologies

*al.*, 2004), une application serveur basée-ontologie a été conçue permettant la conception, le développement et la gestion de composants logiciels. Cette ontologie modélise la description des propriétés, des relations et des comportements des composants logiciels. S'appuyant sur cette ontologie, les composants peuvent être interrogés, pré-chargés, contrôlés et composés pendant le développement aussi bien que pendant l'exécution. Une autre application intéressante des ontologies lors d'un processus MDA est présentée en (Wagelaar, 2008) où les ontologies sont utilisées pour représenter la connaissance sur la diversité des plateformes et comment cette information est utilisée pour exécuter des configurations valides permettant des transformations adéquates entre les modèles de plateformes.

Afin d'améliorer les capacités de déploiement automatique, une approche SOA (Service Oriented Architecture) peut être utilisée pour identifier les meilleurs services qui peuvent être fournis et composés dans le but de satisfaire les exigences demandées. SCA (Service Component Architecture) (Chappell, 2007) est un ensemble de spécifications pour le développement d'applications selon l'approche SOA. L'assemblage de différents composants s'interconnectant par l'offre ou la référence à des services permet une construction des applications qu'elles soient distribuées ou non, s'exécutant sur un ou plusieurs processeurs. Les composants peuvent être inclus dans des composites. Ces derniers offrent alors des services par la promotion de certains services de leurs composants et peuvent avoir besoin de services offerts par d'autres composites. Enfin, un ensemble de composites interconnectés dans la même implémentation SCA forme un domaine. Pour décrire les composites et les composants, SCA utilise le langage SCDL (Service Component Definition Language) formalisé sur le langage XML. L'accès distant à des services est possible par l'utilisation de liaison (ou binding) (e.g. Corba IIOP, web services, etc.). De plus, SCA permet l'implémentation de composants dans beaucoup de langages tels que Java, C++, PHP, JavaScript, BPEL (Barber, 2007).

## 2.2 Standards, protocoles et implémentations multimédia

Ces dernières années, plusieurs organisations telles que l'ITU, le W3C et l'IETF ont fait un gros travail de standardisation lié aux systèmes multimédia. Les paragraphes suivants synthétisent plusieurs de ces standards.

**Recommandation ITU-T F700 (ITU-T F.700).** F.700 propose un cadre de travail pour développer et décrire des services multimédia. D'un point de vue fonctionnel, ce standard fournit une méthodologie pour le développement de services multimédia en considérant les besoins à la fois des utilisateurs finaux et des fournisseurs de service. Cette approche est fondée sur un modèle à quatre niveaux, à savoir *application*, *service*, *tâche de communication* et *composant média*.

Le niveau *application* décrit les caractéristiques fonctionnelles du point de vue de l'utilisateur. Le niveau *service* inclut les services ou les outils qui satisfont les exigences fonctionnelles du niveau *application*. Les services tels que la qualité de service (QoS) la sécurité ou l'intercommunication sont définis et décrits au niveau *service*. Selon le modèle, combiner les tâches de communication et coordonner leurs interactions génèrent la construction des services. Puis, au niveau *tâche de communication*, les tâches sont définies comme des entités fonctionnelles des services multimédia. Ils manipulent des composants multimédia afin de

gérer de l'information. Les fonctions telles que le transfert, le stockage, et l'échange sont définis à ce niveau. Enfin, au niveau le plus bas du modèle, le niveau *composant média* spécifie les aspects multimédia des services en décrivant chaque composant monomédia tels que l'audio, la vidéo, etc., portant l'information de l'utilisateur et en établissant les fonctions telles que la capture, le codage, la présentation, etc., de cette information. Du point de vue des activités de contrôle, la recommandation propose un contrôle et un plan de traitements qui interagissent avec les niveaux *service*, *tâches de communication* et *composant média* par des éléments intergiciels du service.

Dans une approche *top-down*, F.700 suggère la décomposition d'un service multimédia en tâches de communication contrôlées par l'utilisateur et/ou par les fournisseurs de service. Dans une approche *bottom-top*, une tâche de communication peut être vue comme l'assemblage des différents composants média exigés par le service multimédia. Ceci rend leurs descriptions particulièrement importantes. Pour pouvoir représenter ces descriptions, la recommandation propose trois attributs à savoir configuration de la communication, entité de contrôle et flux d'information pour décrire les tâches de communication génériques telles que l'envoi, la conversation, la conférence, la distribution, la collecte et la réception.

En d'autres termes, F.700 donne la possibilité de décrire et de construire des services multimédia de façon automatique, par exemple, en développant une ontologie qui décrit la composition de services multimédia, compréhensible par des machines, permettant de faire de la configuration et/ou la construction automatique de services multimédia. De plus, d'autres recommandations peuvent être incluses avec F.700, dans le but de décrire ou de développer des services multimédia en considérant en même temps, par exemple, des aspects QoS inclus dans la recommandation ITU-T X.461 ou des exigences utilisateur pour le délai et pour la perte d'information décrits dans la recommandation ITU-T G.1010.

**Standards IETF MMUSIC et AVT.** Le groupe de travail MMUSIC<sup>2</sup> de l'IETF a été le responsable d'une spécification des protocoles de session les plus largement utilisés pour des systèmes multimédia. Le protocole de description de sessions (SDP) fournit une représentation commune pour exprimer des descriptions de média et de session. SDP propose un format de données entièrement textuel pour maximiser la portabilité entre applications distribuées. Récemment, une nouvelle version de SDP appelée SDPng a été faite pour augmenter l'interopérabilité et la capacité d'extension en utilisant une syntaxe basée sur XML. D'autres protocoles standardisés par le groupe de travail MMUSIC sont largement connus tels que SIP, RTSP et SAP :

- le protocole d'initiation de session (SIP) est largement utilisé pour des applications interactives (e.g., VoIP, vidéoconférence, etc.) ;
- le protocole de streaming temps réel (RSTP) est utilisé comme le protocole de contrôle de session pour des applications multimédia d'envoi (e.g., IPTV, vidéo à la demande, etc.) ;
- le protocole d'annonce de session (SAP) est quant à lui utilisé pour assister les publications des conférences multimédia multicast et d'autres sessions multicast.

Un autre protocole proposé par l'IETF est le protocole de transport temps réel (RTP). RTP a été fourni par le groupe de travail « Audio Video Transport » comme le protocole

---

<sup>2</sup> Multiparty Multimedia Session Control

MODA : une architecture dirigée par les ontologies

standard pour la représentation de flux multimédia en streaming. RTP est indépendant de tout protocole réseau et transport, aussi il est souvent utilisé au-dessus d'UDP. Actuellement, RTP est utilisé comme le standard pour les services de streaming en réseau à la fois unicast et multicast.

**Plateformes d'implémentation logicielle.** Une grande diversité d'infrastructures multimédia consacrées au développement d'applications multimédia distribuées ont été mises à jour ces dernières années. La plupart ont été implémentées dans des environnements Java, C/C++ et .net. Quelques exemples d'infrastructures de développement sont :

- Java Media Framework API (JMF) : JMF étend la plateforme Java 2 Standard Edition (J2SE) en fournissant une technologie multi-plateformes pour des processus média fondés sur le temps (e.g., capture, play-back, streaming, transcodage entre de multiples formats de média). JMF est un environnement multi-plateformes et peut être utilisé pour implémenter des applications s'exécutant sur les systèmes d'exploitation Windows, Mac OS ou Linux ;
- Java Micro Edition (JavaME) : fournit un environnement standard pour des applications s'exécutant sur des mobiles ou d'autres composants imbriqués (e.g., téléphones mobiles, PDAs, boîtiers décodeurs TV, etc.). L'API Mobile Media (MMAPI) est un module optionnel dans la plateforme Java ME fournissant une API standard pour le calcul de média fondés sur le temps ;
- Android : est une plateforme Java pour les pilotes de mobile incluant un système d'exploitation, une couche intergicielle, des services et des applications clé. L'environnement Android fournit un support pour construire des applications intégrant des fonctions de calcul multimédia ;
- iPhone : le système d'exploitation de l'iPhone propose aussi un environnement pour développer des applications multimédia très riches. Il utilise principalement le langage de programmation Objective-C qui est une extension du langage C ANSI ;
- Autres : des plateformes telles que .net / Windows Mobile OS ont aussi des infrastructures pour développer des applications multimédias.

Cette liste non exhaustive de plateformes logicielles spécifiques montre la diversité des langages et des environnements disponibles aujourd'hui pour développer des systèmes multimédia distribués. Cette diversité des plateformes logicielles associées aux différents standards et protocoles nous a suggéré de proposer une infrastructure pour une architecture multimédia dirigée par les ontologies (dite MODA) pour des systèmes multimédia en réseau.

### 3 L'infrastructure MODA

Dans la section précédente, nous avons présenté les derniers travaux dans les domaines des méthodologies aussi bien que les architectures SCA. D'autre part, les efforts de propositions d'infrastructure pour les standards multimédia aussi bien que la définition de protocoles multimédia à prendre en compte ont été exposés. La diversité des langages et des environnements de développement pour l'implémentation des applications multimédia ont aussi été présentées. Tout ceci nous a amené à concevoir une architecture multimédia dirigée par les ontologies (MODA) dans le but d'automatiser le déploiement et la configuration de systèmes

multimédia en réseau. Nous allons détailler dans cette section les caractéristiques principales de MODA.

Pour prendre en compte les standards bien connus vus précédemment, cette architecture intègre les concepts du multimédia, de la distribution et des plateformes spécifiques dans le but de permettre la traduction dynamique des demandes de l'utilisateur ou d'une communauté d'utilisateurs et la spécification du déploiement des composants. Les spécifications de déploiement générées dans le cadre de MODA peuvent aussi être utilisées pour auto-configurer le système de communication (e.g., le pare-feu, NAT, etc.) et pour déployer de même les composants de communication adéquats dans les couches intergicielles, transport ou réseau afin de fournir la qualité de service requise pour garantir les exigences de la session multimédia. Dans cet article, nous n'exposons que les bases du cadre de travail guidant la génération automatique des spécifications de déploiement. Les paragraphes suivants décrivent les ontologies composant MODA aussi bien que le processus permettant la génération automatique des spécifications de déploiement.

### 3.1 Ontologies intégrées dans MODA

**Ontologie des services multimédia de l'ITU.** Dans ce paragraphe, nous décrivons l'ontologie que nous avons construite à partir de la recommandation F.700. Comme nous l'avons mentionné précédemment, une part importante de F.700 est la description de la construction des services multimédia à partir des tâches de communication. Aussi, l'ontologie F.700 est focalisée sur la description des tâches de communication par un ensemble de classes et de propriétés à savoir :

- *configuration de la communication* (CommunicationTask) : dont le but est d'exprimer si la communication est point-à-point, point-à-multipoint, multipoint-à-point ou multipoint-à-multipoint ;
- *symétrie du flux d'information* (Flowsymmetry) : spécifie la direction dans laquelle l'information est envoyée ;
- *entité de contrôle de la transmission* (TransmissionControlEntity) : cela permet de savoir qui contrôle la transmission de l'information ;
- *délai de communication* (CommunicationDelay) : soit le type de délai supporté par la tâche de communication ;
- *media* : le médium, ou les média, obligatoires ou optionnels, transmis par la tâche de communication. En accord avec la recommandation, les médias ont aussi un niveau de qualité spécifié ;
- *interrelation des médias* : ceci permet de spécifier s'il y a synchronisation ou non entre eux (i.e. synchronisation entre les sous-titres et le mouvement des lèvres), symétrie pour indiquer si le même type de média est bidirectionnel ou la conversion entre média pour indiquer quand un médium est converti dans un autre type de média. Par exemple quand des graphiques sont convertis en images fixes.

Des relations entre ces classes sont définies qui elles-mêmes portent des propriétés mathématiques. Dans la figure 1, nous pouvons voir les relations entre les classes principales de notre ontologie F.700. Par exemple, *hasControl* spécifie qui contrôle la communication, a

MODA : une architecture dirigée par les ontologies

une relation symétrique *hasControlledBy* et est caractérisée par la relation mathématique de disjonction.

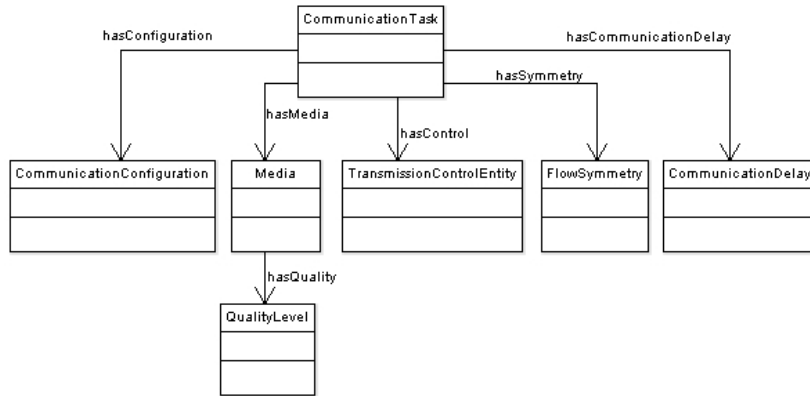


FIG. 1 – Ontologie F.700 des services multi-média de l'ITU.

Dans cette ontologie, les tâches de communication *envoi*, *conversation*, *conférence*, *distribution*, *collecte* et *réception* sont ajoutés comme des sous-classes de la classe *CommunicationTask*. Il est possible de décrire en même temps toutes les relations entre la classe *CommunicationTask* et ses sous-classes. Aussi pouvons nous décrire correctement n'importe quel individu (ou instance) qui correspond à n'importe quelle de ces tâches de communication.

**Ontologie des protocoles de l'IETF.** A partir des divers protocoles standards spécifiés par l'IETF, nous avons conçu de même une ontologie des protocoles multimédia intégrant le contrôle et la description de session et les protocoles de transmission (Figure 2).

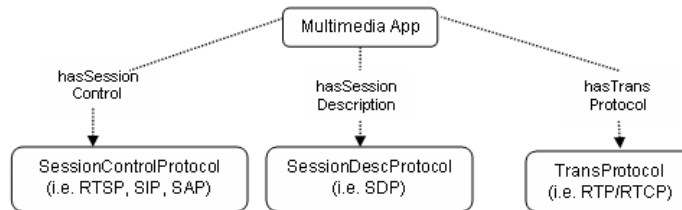


FIG. 2 – Ontologie des protocoles de l'IETF.

Les individus de cette ontologie sont représentés par des applications implémentant les protocoles de contrôle de session RTSP, SIP ou SAP afin de négocier et d'établir la session multimédia requise décrite par l'implémentation des protocoles SDP. Pour la plupart de ces applications, le protocole RTP/RTCP sera implémenté dans le but d'échanger les données multimédia entre les composants du système distribué (e.g., de l'émetteur vers le récepteur pour la VoD).



**Ontologie SCA.** Enfin, nous avons mis en place une ontologie SCA intégrant tous ces concepts, leurs relations et leurs attributs. La figure 3 montre les classes principales de cette ontologie.

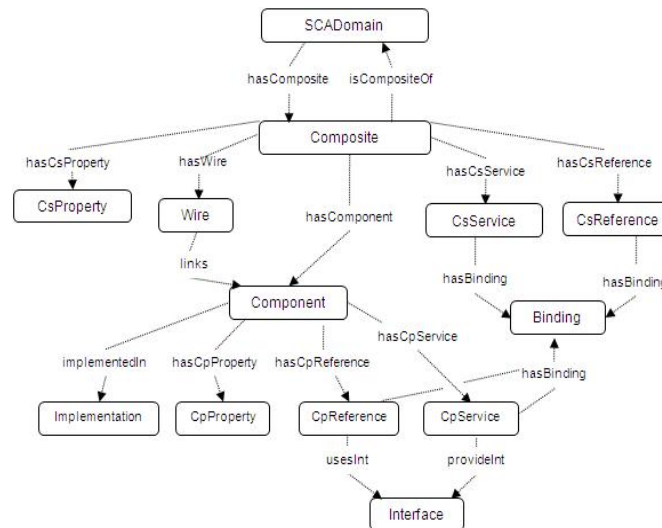


FIG. 3 – *Ontologie de la recommandation SCA.*

**Ontologie des implémentations logicielles.** Dans le but de faciliter la traduction automatique des modèles spécifiques à une plateforme en utilisant une approche dirigée par les ontologies, une ontologie multimédia spécifique plateforme décrivant les composants des systèmes multimédia générique a été élaborée (voir Figure 4).

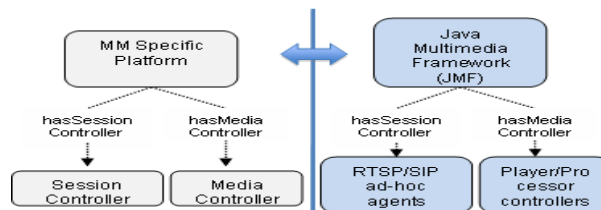


FIG. 4 – *Ontologie des implémentations logicielles*

Cette ontologie définit principalement les deux composants de base qui doivent être déployés sur chaque hôte participant à la session multimédia à savoir le contrôleur de session et le contrôleur de média. Le premier est conçu pour décrire, négocier, établir et terminer la session multimédia entre les participants alors que le second est conçu pour prendre en charge les fonctions liées au multimédia (i.e., capture, codage/décodage, transmission, réception, enregistrement, présentation). Pour incorporer une plateforme spécifique dans le cadre de

MODA : une architecture dirigée par les ontologies

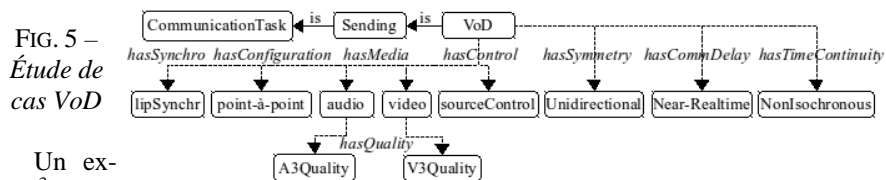
MODA, des individus de cette plateforme spécifique doivent être instanciés. Par exemple, si l'on veut intégrer une implémentation JMF, un individu de JMF lié à un ou plusieurs composants de contrôle de session (e.g., serveur RTSP, client RTSP, agent SIP, etc.) et un ou plusieurs composants de contrôle de média (e.g., processeur, etc.) doivent être ajoutés.

### 3.2 Processus MODA

Le processus MODA inclut les trois niveaux CIM/PIM/PSM préconisés par l'OMG. Dans notre cas, le niveau CIM représente les systèmes multimédia en réseau tenant compte des différentes ontologies des standards. Le niveau PIM utilise le modèle SCA pour obtenir une description générique des systèmes multimédia à partir de la représentation CIM. Au niveau PSM, les ontologies d'implémentation peuvent être utilisées pour traduire notre représentation PIM des systèmes multimédia en réseau. Les étapes du processus MODA pour construire automatiquement un système multimédia en réseau est constituées d'une suite de transformation d'ontologies à partir de règles stipulées en XSL. La première étape consiste en la transformation d'une instance d'une source multimédia OWL par le moteur CIMtoPIM pour produire une instance de l'ontologie SCA. Lors de la seconde étape, le moteur PIMtoPSM prend en paramètre d'entrée une instance SCA et réalise le déploiement des composants pour une plateforme spécifique choisie (e.g., JMF ou JavaME) aussi selon des règles définies.

## 4 Étude de cas : Vidéo à la demande (VoD)

Nous avons évalué MODA pour le déploiement automatique d'un système VoD. Dans le but de caractériser un tel type d'application, les ontologies MODA ont été utilisées pour décrire les tâches de communication impliquées. L'individu décrivant notre application VoD et sa tâche de communication « *Sending* » est présenté dans la figure 5.



Un ex-  
trait<sup>3</sup> des

règles de transformation XSL entre les ontologies multimédia et SCA sont décrites dans la figure 6 ci-dessous qui pour une configuration décrite en F700 génère les composants et composites SCA à mettre en place :

<sup>3</sup> Par manque de place, un exemple simple des ces règles est inclus dans cet article (voir [www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/Main pour le détail des règles](http://www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/Main_pour_le_d%C3%A9tail_des_r%C3%A8gles))

```

0. ...
1. <xsl:template name="hasCommunicationConfiguration">
2.   <xsl:element name="Component">
3.     <xsl:element name="hasImplementation">Java</xsl:element>
4.     <xsl:choose>
5.       ...
6.       <xsl:when test="./local-name()='PointToPoint'">
7.         <xsl:attribute name="name">
8.           <xsl:value-of select="./PointToPoint/@ID"/>
9.         </xsl:attribute>
10.        <xsl:element name="hasReference">.....</xsl:element>
11.      </xsl:when>
12.    </xsl:choose>
13.  </xsl:element>
14. ....
15. </xsl:template>
16. ...

```

FIG. 6 – Règles de transformation XSL

Une tâche de communication génère un individu SCA pour l'émetteur et les récepteurs. Puis, pour chaque participant, les règles de transformation spécifient la génération des différents composants. Une tâche de communication génère un domaine SCA pour l'émetteur et les récepteurs. Puis, pour chaque domaine généré, les composants et les composites sont construits. Par exemple, le moteur CIMtoPIM produit pour la balise *<hasCommunicationConfiguration>* de l'individu F700 (ligne 1) un nouveau composant (ligne 2) pour un composite du domaine de l'émetteur VoD avec une implémentation future en Java (ligne 3). Si *<hasCommunicationConfiguration>* contient la balise *<PointToPoint>* (ligne 6), le nom du nouveau composant est la valeur de l'attribut ID de cette balise (ligne 7 à 9) et ce composant référence un service (ligne 10) d'un autre composant (dans notre exemple, le domaine du récepteur VoD)<sup>4</sup>.

## 5 Conclusions et perspectives

Dans cet article, nous avons présenté MODA qui est un cadre de travail pour le déploiement et la configuration automatique de systèmes multimédia en réseau. MODA intègre un ensemble d'ontologies concernant les standards les plus connus dans le domaine du multimédia, de la distribution et les concepts liés à des plateformes spécifiques. De plus, il fournit un processus automatique pour guider la génération dynamique des configurations de déploiement des systèmes multimédia. L'utilisation de MODA pour permettre la traduction automatique entre les demandes des utilisateurs et la spécification du déploiement des composants a été illustrée par l'instanciation du système VoD. MODA est encore un prototype et son développement est encore en cours dans le cadre du projet Feel@Home (Feel, 2009) mais les résultats préliminaires sont très encourageants pour déployer automatiquement des services multimédia pour une communauté d'utilisateurs. Les travaux futurs vont concerner la génération dynamique des interfaces utilisateur pour les usages finaux (pour conduire le processus

<sup>4</sup> Dû à la limitation de l'espace, de plus amples informations sont données à : [www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/Main](http://www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/Main)

MODA : une architecture dirigée par les ontologies

de déploiement) et la validation des spécifications de déploiement produites par MODA pour être utilisés lors de l'auto-configuration du système de communication (e.g., pare-feu, NAT, etc.). Nous allons aussi essayer de déployer des composants de communication adéquats dans les couches intergicielle, transport ou réseau pour fournir la qualité de service requise garantissant ainsi les exigences de la session multimédia.

## Références

Barber G. (2007], *What is SCA*, <http://www.osoa.org>

Bezivin, J. (2004). *Sur les principes de base de l'ingénierie des modèles*, journal RSTI-L'objet ISSN 1262-1131, vol. 10, n° 4, pp. 145-157.

Chappell, D. (2007), *Introducing SCA*, <http://www.davidchappell.com>

Feel (2009). Site web du projet Feel@home, <https://rd-projetfeelathome.rd.francetelecom.com/public>

Poole, J. D. (2001). *Model-Driven Architecture: Vision, Standards and Emerging Technologies*, Workshop on Metamodeling and Adaptive Object Models, ECOOP.

Miller, J., et J. Mukerji (2003). *MDA Guide Version 1.0.1*, n° document : omg/2003-06-01

Oberle, D., et A. Eberhart, S. Staab, R. Volz (2004). *Developping and Managing Software Components in an Ontology-based Application Server*, ACM/IFIP/USENIX, 5<sup>th</sup> International Middleware Conference, Toronto, Canada, volume 3231 of LNCS, p. 459-477.

ODA (2001). *Ontology Driven Architectures and Potential Uses of the semantic Web in Systems and Software Engineering*, <http://www.w3.org/2001/sw/BestPractices/SE/ODA/>

Wagelaar, D. (2008). *Platform Ontologies for the Model-Driven Architecture*, thèse de doctorat, Faculté des sciences de l'université Vrije Bruxelles.

MODA, (2009). *Multimedia Ontology Driven Architecture framework*, <http://www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/Main>

## Summary

Heterogeneous multimedia environments make difficult the automatic deployment and interoperability of networked multimedia systems. Diversity of languages, protocols, and hardware platforms lead to major incompatibility issues. Moreover, dynamic multimedia systems instantiation and configuration guided by user and community requirements and preferences are not easy to be performed in this context. This work introduces an ontology framework for automatic networked multimedia systems deployment and configuration. Based on well-known standards, this framework integrates multimedia, distribution and plat-

M. Lamolle *et al.*

form concepts. A study case illustrating how networked multimedia systems can be dynamically deployed is presented in order to demonstrate the advantages of using this framework.