

Vers l'interopérabilité des Systèmes de DRM (Digital Rights Management)*

Majirus Fansi Vincent Lalanne Alban Gabillon

Université de Pau et des Pays de l'Adour. IUT des Pays de l'Adour, 371 Rue du Ruisseau, 40004 Mont de Marsan, France.

E-Mails : janvier-majirus.fansi@etud.univ-pau.fr, [[vincent.lalanne](mailto:vincent.lalanne@univ-pau.fr), [alban.gabillon](mailto:alban.gabillon@univ-pau.fr)] @univ-pau.fr

Un des principaux inconvénients des systèmes de gestion des droits numériques actuels est le manque d'interopérabilité entre ces systèmes. Dans cet article, nous identifions les obstacles à l'interopérabilité des solutions DRM et proposons des solutions alternatives. Ainsi, nous préconisons l'usage des langages d'expression de droits ODRL et MPEG-21 REL. Aussi, nous proposons une interface de service web pour chaque acteur de l'architecture DRM afin de faciliter les échanges entre différents systèmes. Les différents composants du système DRM communiquent alors par échanges de messages SOAP. Nous tirons parti de la propriété d'extensibilité de XML pour véhiculer avec le contenu protégé, les informations nécessaires à son rendu par tout logiciel client sous réserve que l'utilisateur détienne une licence.

Mots Clés: DRM, XML, XrML, MPEG-REL, ODRL, Web Services, Contrôle d'Usage, Sécurité.

1. Introduction

La distribution de contenus numériques via Internet offre de nouveaux services aux consommateurs et fournit de nouvelles opportunités commerciales aux créateurs de contenu. Cependant le succès de ces nouveaux canaux de distribution dépend de l'effectivité des mécanismes de protection des intérêts des différents intervenants de la chaîne. Ces mécanismes sont assurés par les systèmes de gestion des droits numériques ou systèmes DRM (Digital Right Management). Les systèmes DRM ont donc pour but de contrôler et de protéger la propriété intellectuelle des contenus numériques tels que les documents, les images, les vidéos et les sons. Ainsi ils restreignent au travers de licences d'utilisation, ce que l'utilisateur peut faire du contenu qu'il a acheté. Une licence spécifie un règlement énonçant les conditions d'utilisation d'un contenu numérique. A ce jour plusieurs systèmes de gestion de droits numériques existent sur le marché. La majorité de ces systèmes, comme Apple iTunes, utilise une solution de type logiciel. Il existe aussi des solutions de type matériel telles que le système DVD-CSS (Content Scramble System).

Dans cet article nous nous intéressons aux solutions logicielles. Chacun de ces systèmes emploie une technique propriétaire de protection de contenus. Les licences délivrées par un système ne sont interprétables que par lui seul. Cette situation a conduit à la mise sur le marché de produits qui ont pour but de faciliter l'interopérabilité des systèmes existants. Nous pouvons prendre comme exemples les Systèmes DMDfusion et OPERA. DMDfusion incorpore les systèmes Microsoft, Adobe et Real Networks. Le Système OPERA quant à lui définit une architecture pour l'interopérabilité des systèmes DRM. Cependant cette architecture ne permet pas d'échanges entre systèmes de DRM puisque les licences sont éditées pour un système particulier et ne sont pas interprétables par les autres systèmes. Sur le plan législatif le besoin d'interopérabilité se fait aussi ressentir. C'est dans ce sens que la loi Française n° 2006-961 du 1^{er} août 2006, dite loi DADVSI, relative au droit d'auteur et aux droits voisins dans la société de l'information dispose à son article L. 331-5 que: les mesures

* Réalisé dans le cadre du projet CASC (ACI Sécurité Informatique 2003-2006), avec le soutien du Conseil Général des Landes

techniques ne doivent pas avoir pour effet d'empêcher la mise en œuvre effective de l'interopérabilité, dans le respect du droit d'auteur. Les fournisseurs de mesures techniques donnent l'accès aux informations essentielles à l'interopérabilité.

Le but de cet article est d'identifier les obstacles à l'interopérabilité des systèmes DRM et de proposer des solutions alternatives. Nos propositions sont destinées à faciliter l'interopérabilité des différents composants qui interviennent dans l'architecture DRM. Il s'agit pour nous d'exploiter les standards existants pour permettre à ces composants de communiquer entre eux afin de rendre de façon efficace et sécurisée le service de DRM.

Concrètement l'interopérabilité des systèmes de DRM dépend:

- du langage d'expression de droits (REL - Rights Expression Language) utilisé dans chacun des systèmes. En effet, la licence d'utilisation est éditée dans un langage et n'est interprétable que par les systèmes qui utilisent le même langage.
- des protocoles de communication entre les composants du système. Les systèmes ne peuvent pas communiquer si chacun utilise des protocoles propriétaires et compréhensibles que par lui seul.
- du mécanisme de protection du contenu. Chaque système doit véhiculer les informations nécessaires au rendu du contenu sécurisé par l'utilisateur ayant acquis une licence. Par exemple, le logiciel client doit connaître l'algorithme utilisé pour protéger le contenu afin de le décrypter.

Les principales contributions de ce papier sont les suivantes:

- Nous montrons l'intérêt de l'utilisation des langages ODRL [4] et MPEG-21 REL [5] pour l'édition des licences (section 4).
- Nous tirons parti du standard OASIS Web Services Security: SOAP Message security pour l'échange de messages entre les acteurs d'une architecture DRM (section 5). Nous proposons alors une interface de services web (Web Services) pour chaque composant du système au travers des squelettes de messages émis et attendus. Chaque élément de l'ensemble peut retrouver un service dans le système à partir de la définition d'interface du composant qui le fournit. Les différents composants communiquent par échanges de messages SOAP (Service Oriented Architecture Protocol).
- Nous exploitons la propriété d'extensibilité de XML (eXtensible Markup Language), pour convoyer avec le contenu sécurisé les informations nécessaires à son rendu (section 5).

La section 2 présente les notions préliminaires dont nous aurons besoin. La section 3 expose les principaux acteurs d'un système DRM basique, son architecture de sécurité, ainsi que le processus d'utilisation des ressources protégées. Nous présentons l'état de l'art à la section 6. Finalement la section 7 conclut ce papier.

2. Notions préliminaires

Dans cette section nous présentons les notions qui seront employées tout au long de cet article. Nous nous intéressons en particulier aux recommandations W3C (World Wide Web Consortium) SOAP (Service Oriented Architecture Protocol) et Service Web (Web service).

2.1. Service Oriented Architecture Protocol (SOAP)[6]

SOAP est un protocole de communication basé sur XML (eXtensible Markup Language) qui permet aux applications d'échanger des informations à travers Internet. Avant l'avènement de SOAP les applications communiquaient essentiellement par des appels de procédures à distance RPC (Remote Procedure Call) entre objets telles que DCOM (Distributed Component Object Model), CORBA (Common Object Request Broker Architecture), Java RMI (Remote Method Invocation). Cependant les RPCs posent un problème de sécurité puisque ces types de trafics sont souvent bloqués par les pare feux (firewall) et les serveurs proxy. Un meilleur moyen de communication entre applications consiste à utiliser http (Hyper Text Transfert Protocol). SOAP a été créé pour permettre cela. Il permet aux applications installées sur différents systèmes d'exploitation (SE), avec des

Interopérabilité des systèmes DRM

technologies et langages de programmation différents de communiquer en s'échangeant des messages SOAP (codés en XML) et en utilisant le protocole http. Un message SOAP est un document XML ordinaire contenant les éléments suivants:

- Un élément enveloppe (*Envelope*) obligatoire qui identifie le document XML comme un message SOAP
- Une entête optionnelle (*Header*) qui contient des informations liées au message. Cet élément est généralement utilisé pour convoyer les informations de sécurité liées au message.
- Un élément corps du message (*Body*) qui contient les informations d'appel et de réponse des procédures distantes.
- Optionnellement un élément *Fault* qui fournit les informations à propos des éventuelles erreurs survenues lors du traitement du message.

La figure 1 (a) montre le squelette d'un message SOAP; pour des raisons de simplicité, nous ne faisons pas apparaître les définitions d'espaces nominaux [7].

2.2. Service Web (Web service)

Un service web² est un système logiciel conçu pour permettre l'interopérabilité des interactions entre machines à travers le réseau. Il a une interface décrite dans un format compréhensible par la machine (en l'occurrence WSDL – Web Service Description Language). Les autres systèmes communiquent avec le service web conformément à sa description en utilisant les messages SOAP typiquement convoyés via http. Un service web est une notion abstraite qui doit être réalisé par des agents concrets. Un agent est le dispositif concret (logiciel ou matériel) qui envoie et reçoit les messages, alors que le service est la ressource caractérisée par l'ensemble abstrait des fonctionnalités qui est fourni. Cette distinction s'illustre par le fait qu'un même service web peut être implémenté dans plusieurs langages de programmation différents. Chaque implémentation représente alors un agent, le service rendu restant le même.

Hormis SOAP, les deux autres technologies qui sous-tendent les services web sont UDDI (Universal, Description, Discovery and Integration) et WSDL.

Un document WSDL est au format XML. Il décrit un service web. Il définit la syntaxe et la sémantique des messages et spécifie les méthodes que le service expose, ainsi que les emplacements réseaux où les agents fournisseurs³ peuvent être invoqués.

UDDI est un service de répertoire basé sur XML qui permet à des organisations de publier et de retrouver des services web. UDDI est interrogé par des messages SOAP et fournit l'accès aux documents WSDL.

Un exemple illustratif de ces concepts est le scénario suivant: une organisation met à disposition un standard UDDI pour la distribution de contenus numériques sécurisés. Les entreprises spécialisées dans la protection de contenu (*packager*) publient leurs services dans le répertoire. Les compagnies de distribution de contenus sécurisés (la FNAC par exemple) peuvent alors chercher dans le répertoire l'interface (document WSDL) des *packagers*. Quand l'interface est trouvée, le distributeur communique avec le service immédiatement et reçoit les ressources souhaitées.

Les services web se sont imposés comme solution effective pour l'interopérabilité entre les machines du réseau. Cependant il s'est posé le problème de confidentialité et d'intégrité des messages échangés. Le standard OASIS Web Services Security (WSS)[8] a été proposé pour répondre à ce besoin. WSS étend les messages SOAP afin d'assurer l'intégrité et la confidentialité des messages échangés. L'extension consiste à ajouter les informations de sécurité dans l'entête d'un message SOAP classique.

L'intégrité est assurée par la recommandation XML Signature [9]. XML signature est une recommandation du W3C destinée à permettre l'utilisation de signatures numériques dans les

² www.w3.org/2002/ws/

³ Un agent fournisseur d'un service est un ensemble logiciel qui rend ce service

documents XML. Une caractéristique importante apportée par XML est la possibilité de ne signer que des portions spécifiques d'un même document.

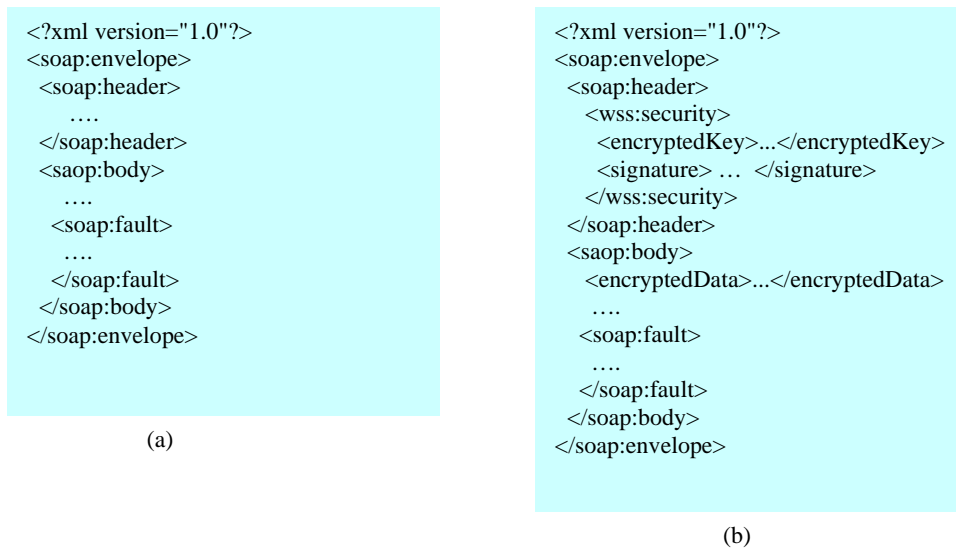


Figure 1 – (a) Squelette des messages SOAP (b) Squelette SOAP + Sécurité

La confidentialité des données transmises est assurée par la recommandation XML Encryption [10]. XML Encryption est une recommandation W3C qui permet de chiffrer les données et de représenter le résultat du chiffrement en XML. Les données à chiffrer peuvent être de toute sorte (y compris un document, une chanson, une vidéo).

La figure 1 (b) montre un message SOAP étendu. L'élément `encryptedData` contient les données chiffrées ainsi que l'algorithme utilisé pour le chiffrement. Il s'agit de cryptographie à clé secrète. La clé utilisée pour le chiffrement est à son tour chiffrée par la clé publique du destinataire. Le résultat est convoyé par l'élément `encryptedKey` qui renseigne aussi sur l'algorithme utilisé pour protéger la clé secrète. Enfin l'élément `signature` contient la signature numérique de l'expéditeur. Il renseigne sur la portion du message concernée par la signature, la clé à utiliser pour la valider et sur l'algorithme utilisé.

3. Structure des systèmes DRM

Cette section présente la structure basique des systèmes DRM. Elle montre comment les différents acteurs sont agencés pour accéder aux ressources protégées et définit les types de contrôleurs DRM.

3.1. Principaux Acteurs des systèmes DRM

Bartolini et al. [1] définissent l'ensemble des besoins des systèmes de gestion de contenu en terme d'ensemble d'acteurs. Chacun de ces acteurs jouent différents rôles dans le système DRM. Les rôles décrits par Bartolini et al. sont:

1. l'auteur (ou le créateur) est responsable de la création du contenu.
2. le détenteur de droit est celui qui détient le copyright du contenu. Il est référencé dans la majorité des systèmes DRM comme le `content owner`. L'auteur n'est pas nécessairement le détenteur du copyright.
3. le producteur est l'entité responsable de la protection de la ressource. Il est encore appelé `packager` dans les systèmes DRM.

Interopérabilité des systèmes DRM

4. le distributeur est l'entité responsable de la distribution du contenu protégé. Il correspond à la composante `content distribution` des systèmes DRM.
 5. le registre IPR (Intellectual Property Rights) est l'entité qui délivre des licences aux utilisateurs. Il est généralement appelé `license server` dans les systèmes DRM.
 6. le générateur d'identifiant ou UNI (Unique Number Issuer) génère un identifiant unique pour chaque création. Ce service est rendu par le `packager` dans les systèmes DRM.
- Ces acteurs constituent avec le `client DRM` l'architecture basique des systèmes DRM que nous représentons à la figure 2.

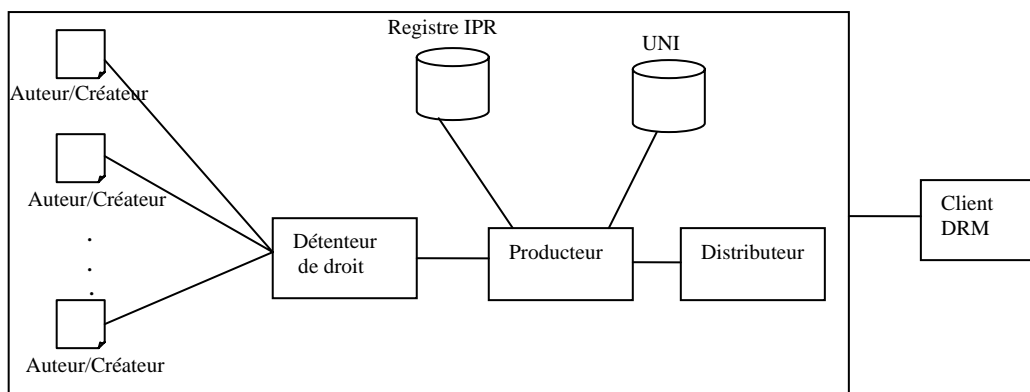


Figure 2 – Acteurs dans un Système DRM

3.2. Architectures de sécurité

Park et al.[2] énoncent trois facteurs qui distinguent les différentes architectures de sécurité impliquées dans la distribution des contenus sécurisés: la présence de la machine virtuelle, l'ensemble des règles de contrôle et le mode de distribution.

Le premier niveau de distinction est la présence de la machine virtuelle. Elle est décrite par Park et al.[2] comme étant "un logiciel qui tourne au dessus d'un environnement informatique vulnérable et emploie des fonctions de contrôle pour fournir les moyens de protéger et de gérer l'accès et l'usage de la ressource numérique". La machine virtuelle est généralement incorporée dans les systèmes DRM sous forme de `plugin`. Les clients qui ne possèdent pas une telle machine virtuelle ne peuvent pas gérer et contrôler l'accès et l'usage aux données sécurisées. Aujourd'hui, chaque système DRM possède sa propre machine virtuelle propriétaire. Par exemple, les morceaux protégés par le système Apple `FairPlay` ne peuvent être écoutés que via `iTunes music player`. De même ceux protégés par `Windows DRM` ne peuvent être écoutés que via la machine virtuelle `Windows Media Rights Manager` pluggée au logiciel `Windows Media Player`.

Le second niveau de distinction concerne l'ensemble des règles de contrôle. Les règles de contrôle d'accès et d'usage de contenus numériques sont exprimées par les langages d'expression de droits (REL). Ces règles ainsi exprimées représentent la licence d'utilisation. Park et al. distinguent trois façons de distribuer la licence:

- licence prédéfinie ou fixe: la machine virtuelle est équipée de la licence d'utilisation pour toutes les ressources qu'elle est censée contrôler. Ce mode de distribution est facile à implanter mais présente l'inconvénient d'être très peu flexible. Le système de chiffrement pour DVDs (`DVD-CSS`) est un exemple de contrôle par licence prédéfinie.
- Licence encastrée: la licence est encastrée dans la ressource protégée. Ceci peut être fait en insérant la ressource et la licence dans une enveloppe de sécurité.
- Licence séparée: la licence et la ressource sont distribuées séparément.

Plusieurs systèmes DRM existant utilisent les deux derniers modes de distribution de licences. Par exemple, Apple `iTunes Music Store` utilise les licences encastrées (`Apple FairPlay`) combinées à une licence fixe basique incorporée dans `iTunes music player`.

Le troisième et dernier niveau de distinction repose sur le mode de distribution du contenu protégé. En effet, l'utilisateur peut obtenir la ressource par un canal direct tel que la messagerie électronique. Il peut aussi le recevoir par téléchargement via un serveur de distribution.

3.3. Utilisation des ressources protégées

La figure 3 montre comment Erickson [3] décrit l'usage des contenus numériques protégés dans la majorité des systèmes DRM. Ces systèmes utilisent pour la plupart des licences séparées. Le contenu est d'abord protégé par le packager et distribué aux utilisateurs. Un serveur de licence (license server) crée une licence d'utilisation à partir des droits de l'utilisateur. La machine virtuelle (DRM Controller) se sert de la licence pour rendre le contenu à l'utilisateur. Différentes solutions DRM utilisent des techniques différentes pour produire le contenu sécurisé (content package), des formats de licences (license) différents et des protocoles de communication entre composants (Content packager, License server, Client) différents.

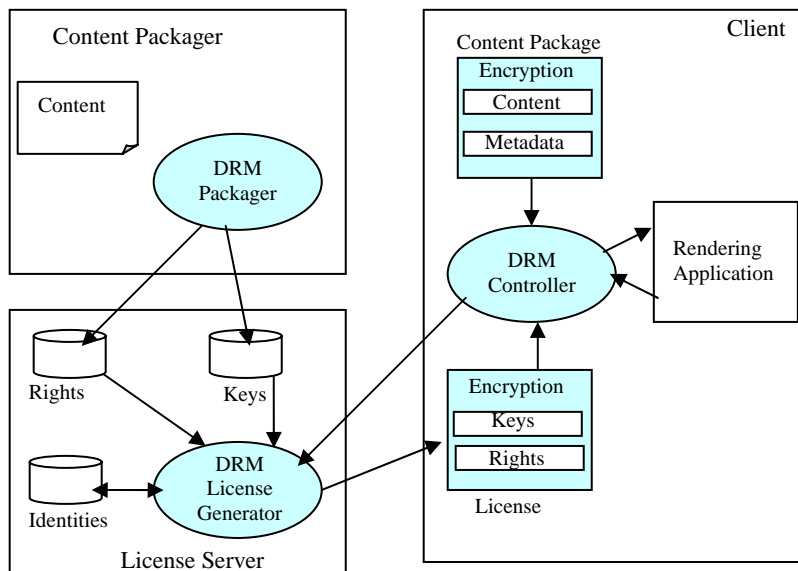


Figure 3 – Usage d'une ressource DRM

4. Langages d'Expression de Droits:REL (Rights Expression Languages)

L'effectivité de l'interopérabilité des systèmes de DRM dépend en partie du langage d'expression des droits utilisé pour créer les licences dans les différentes solutions. En effet, afin que deux systèmes se comprennent il est nécessaire que la licence éditée avec un des systèmes soit compréhensible par l'autre et vice versa. Un standard REL est donc l'une des clés pour l'interopérabilité des systèmes DRM propriétaires.

Plusieurs langages de droits ont été proposés XrML⁴, ODRL [4], MPEG-21 REL [5], FORM [11]. A ce jour, deux principaux langages permettent de décrire les licences. D'un côté, ODRL (Open Digital Rights Language) est une recommandation W3C. Un sous ensemble de ODRL est utilisé par le forum des industries leaders du mobile (OMA - Open Mobile Alliance) dans le cadre du système DRM OMA. De l'autre côté, MPEG-21 REL est un standard ISO/IEC. Dans cette section,

⁴ www.xrml.org

Interopérabilité des systèmes DRM

nous décrivons brièvement les langages MPEG-21 REL et ODRL. Ensuite nous les positionnons par rapport à l'objectif d'interopérabilité.

4.1. MPEG-21 REL

MPEG (Moving Picture Experts Group), est le groupe de travail SC 29/WG 11 de l'ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) pour les technologies de l'information chargé du développement des normes internationales pour la compression, la décompression, le traitement et le codage de la vidéo, de l'audio et de leur combinaison, de façon à satisfaire un large panel d'applications. Un des standards produits par MPEG est MPEG-21. Son but est d'offrir un cadre normatif pour la distribution et l'usage des contenus multimédias à tous les intervenants de la chaîne. MPEG-21 est constitué de plusieurs parties dont la partie 5 qui spécifie le langage d'expression de droits MPEG-21 REL [5].

MPEG-21 REL est basé sur la proposition XrML (eXtensible rights Markup Language). Avec MPEG-21 REL il est possible de spécifier pour une ressource numérique (contenu ou logiciel), qui est autorisé à utiliser cette ressource, les droits disponibles et les conditions nécessaires à l'exercice de ces droits sur la ressource.

Le standard MPEG-21 dans sa partie 6 spécifie un dictionnaire de données qui définit de façon unique les différents termes utilisés par MPEG-21 REL pour exprimer les droits.

Le noyau de MPEG-21 REL est constitué des éléments principal, resource, right et condition (voir figure 4):

- **Principal**: identifie une entité telle que la personne, l'organisation, ou le dispositif à qui les droits sont attribués.
- **Right**: spécifie l'action que l'entité identifiée par l'élément principal peut être autorisé à exercer sur une quelconque ressource.
- **Resource**: identifie l'objet sur lequel est basée l'autorisation. Un URI (Unified Resource Identifier) peut être utilisé pour identifier la ressource.
- **Condition**: spécifie une ou plusieurs conditions qui doivent être satisfaites pour que les droits soient exercés. Par exemple, une condition pourrait limiter l'écoute d'une chanson à un nombre de fois ou pendant un intervalle de temps précis.

Ces éléments sont encapsulés dans un élément grant, lui aussi partie d'un élément englobant license. Une licence (élément license) peut contenir une ou plusieurs autorisations (éléments grant), l'émetteur de la licence (élément issuer), accorde les privilèges contenus dans la licence. La figure 4 (a) montre une structure simple de licence MPEG-21 REL. Comme le montre cette figure, l'émetteur de la licence peut y attacher une signature numérique.

L'élément grant est la partie d'une licence MPEG-21 REL qui accorde à l'entité identifiée par l'élément principal le droit d'utiliser la ressource sous certaines conditions. Par exemple, considérons un fichier audio distribué à un utilisateur (Toto). Le document MPEG-21 REL possède une entrée qui exprime le fait que Toto a le droit d'écouter 3 fois ce fichier audio. La figure 4 (b) montre l'élément grant qui spécifie cette règle.

4.2. ODRL

ODRL est une proposition de standard pour l'expression des droits sur les contenus numériques. ODRL est destiné à fournir des mécanismes flexibles et interopérables pour la publication, la distribution et l'usage transparent des ressources numériques telles que la musique, la vidéo, les logiciels et autres créations numériques. Ce langage est une recommandation du W3C. Il ne nécessite pas de licence et est disponible librement.

Les licences ODRL exploitent un dictionnaire de données formé des éléments qui permettent d'exprimer des droits sur les contenus numériques.

ODRL est basé sur un modèle extensible d'expression de droits qui comprend trois principales entités:

- **Party**: comprend les utilisateurs finaux ou les détenteurs de droit. Les détenteurs de droits sont les entités qui participent à la création, à la production, ou à la distribution des ressources numériques. Cet élément est analogue aux éléments `principal` et `issuer` du standard MPEG-21 REL.
- **Permission**: représente les autorisations, qui peuvent contenir des contraintes, des obligations et des conditions. Les contraintes sont les limites imposées à l'usage de la ressource (par exemple regarder une vidéo au maximum 5 fois). Les obligations sont des pré requis pour l'obtention des permissions (par exemple payer 5\$ chaque fois afin de regarder la vidéo). Les conditions spécifient des exceptions, qui si elles sont satisfaites, révoquent les permissions ou entraînent la renégociation de celles-ci (par exemple si la carte de crédit expire alors toutes les permissions sur la vidéo sont révoquées).
- **Asset**: c'est la ressource à protéger. Elle doit être identifiée de façon unique. C'est l'équivalent de l'élément `resource` du standard MPEG-21 REL.

Considérons l'exemple précédent exprimé avec MPEG-21 REL à la figure 4 (b). La figure 5 donne sa représentation en ODRL en utilisant les entités `party`, `permission` et `asset`.

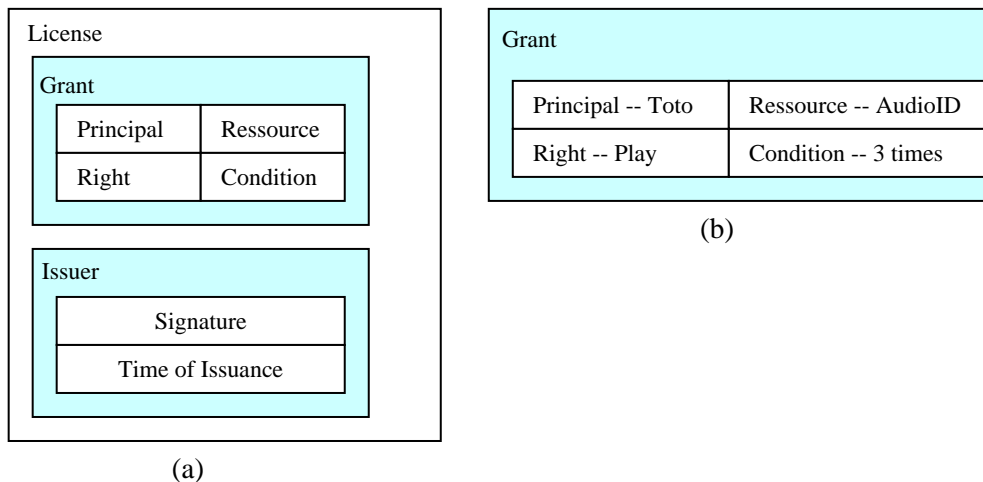


Figure 4 – (a) licence MPEG-21 REL – (b) Autorisation MPEG-21 REL

4.3. Interopérabilité entre MPEG-21 REL et ODRL

ODRL et MPEG-21 REL ont beaucoup de similarités. Ils sont syntaxiquement basés sur XML. Et bien que les termes utilisés dans ces langages soient différents, ils sont sémantiquement très proches et permettent d'adresser à quelques différences près les mêmes besoins. Cependant, il est difficile de faire le choix d'un langage au détriment de l'autre pour deux raisons au moins:

- Les deux langages ODRL et MPEG-21 REL bénéficient respectivement du soutien des organisations de standardisation W3C et ISO/IEC: D'une part, ODRL est une recommandation W3C dont un sous ensemble est utilisé par le forum des industries leaders du mobile (OMA - Open Mobile Alliance) dans le cadre du système DRM OMA. D'autre part, MPEG-21 REL est un standard ISO/IEC. Il est basé sur XrML qui a inspiré certains systèmes DRM propriétaires existants (Windows media Right manager de Microsoft en est un exemple).

Interopérabilité des systèmes DRM

- Ces deux langages sont au format XML. Ils sont donc extensibles et peuvent chacun de son côté évoluer par enrichissement de fonctionnalités. Nous ne pouvons donc dire que l'un est plus expressif que l'autre.

Ces deux langages d'expression de droits sont largement utilisés. Il est donc important de faciliter l'interopérabilité entre les systèmes qui les utilisent. Toute solution DRM doit donc être capable d'interpréter les licences éditées à partir des langages ODRL et MPEG-21 REL.

Dans ce papier nous préconisons l'interprétation des deux langages ODRL et MPEG-21 REL dans les solutions DRM. Le module d'interprétation de licences des clients DRM comporterait alors deux unités, une pour chacun des deux langages. Ainsi à la réception d'une licence, le client DRM utiliserait l'unité appropriée pour l'interpréter.

Party -- Toto		Asset -- AudioID	
Permission --Play		Constraint --3 times	

Figure 5 – Autorisation en ODRL

5. Proposition d'architecture pour des systèmes DRM interopérables

Dans cette section, nous proposons une architecture pour faciliter l'interopérabilité des systèmes DRM. Nous décrivons d'abord les différents composants de l'architecture, avant de détailler l'interface et le protocole de communication entre ces composants.

5.1. Composants de l'architecture

Nous proposons une architecture composée des éléments de base communs à la majorité des systèmes DRM propriétaires existants. Nous exploitons, pour ne citer que quelques unes, les architectures des systèmes Windows Media Rights Manager de Microsoft, Electronic Media Management System de IBM, Media Commerce Suite de RealNetworks. La figure 6 présente notre architecture. Les composants (ou acteurs) de cette architecture sont décrits comme suit:

5.1.1. content owner

Ce composant fournit le contenu numérique, ainsi que les contrats et les droits au packager. Les contrats et les droits détermineront l'usage autorisé du contenu par un utilisateur donné.

5.1.2. packager

Le packager assure les fonctions suivantes: la compression si possible des données et la protection du contenu. La protection du contenu comprend le chiffrement et le watermarking.

Le packager reçoit en entrée la ressource non protégée du content owner. Il fournit les clés de décryptage, l'algorithme utilisé pour protéger le contenu ainsi que les droits et les contrats au composant license/key_distribution. Enfin le contenu protégé est transmis à l'élément content_distribution.

Le watermarking consiste à rajouter, sur un medium (qui peut être une image, une chanson, un film vidéo), une marque (en anglais, watermark signifie filigrane) qui doit être suffisamment imperceptible pour ne pas détériorer le medium et suffisamment robuste pour pouvoir être décelée même après traitement du medium. Le contenu d'une marque, typiquement quelques bits d'information, peut être de différentes natures.

- La marque peut contenir des informations sur les permissions attachées au document. Ces informations reflètent les droits et contrats spécifiés par le content owner. C'est au dispositif terminal client que revient la charge de détecter et de respecter les instructions de la marque en fonction de la licence acquise. Aussi, le client ne doit pas être corrompu.
- La marque peut indiquer l'identifiant du content owner, celui du contenu et l'URL (Uniform Ressource Locator) du serveur de licences.

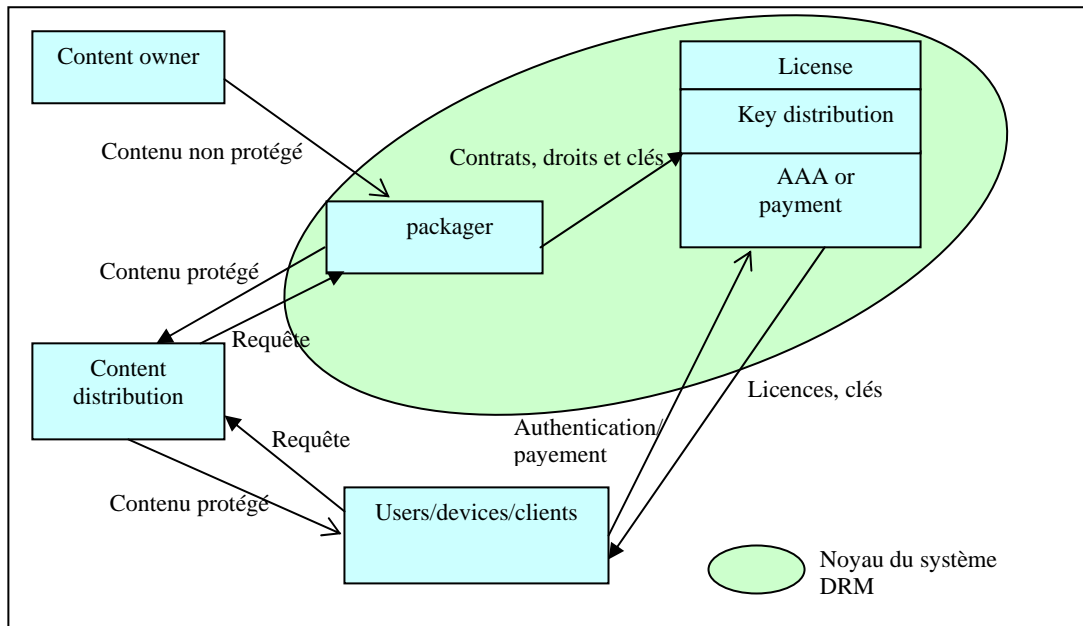


Figure 6 – Architecture DRM

5.1.3. License/key distribution/AAA or Payment

L'entité `AAA_or_Payment` assure l'autorisation, l'authentification et le contrôle d'accès. En complément ou tout simplement à la place de ce processus, les utilisateurs payent les frais de licence. Dès que cette étape est faite, il contacte le module `license/key_distribution` qui fournit la licence à l'utilisateur.

La composante `license/key_distribution` reçoit les droits, contrats et clés de décryptage du `packager`. Il délivre les licences, clés de décryptage ainsi que l'algorithme utilisé pour protéger le contenu aux utilisateurs sur la base:

- des informations fournies par la composante `AAA_or_payment`; Ces informations concernent l'identifiant du contenu et les contraintes d'usages telles que le nombre de fois où l'utilisateur pourra écouter la chanson ou la limite dans le temps de la validité de la licence.
- des contrats et des droits reçus du `packager`.

5.1.4. Content Distribution

Ce composant peut distribuer le contenu sous plusieurs formes

- La distribution peut se faire par Internet en téléchargement ou en streaming (diffusion des contenus audio ou vidéo en continu, au fur et à mesure du téléchargement du fichier);
- Elle peut aussi se faire sur supports physiques tels que DVDs, CDs.

Interopérabilité des systèmes DRM

Il reçoit les contenus sécurisés du `packager`. Chaque contenu à un identifiant qui permet d'aller chercher la licence et la clé de décryptage.

5.1.5. Users/devices/clients

Les droits peuvent être associés à l'utilisateur ou au matériel. Une application cliente est responsable du contrôle d'usage et du décryptage du contenu numérique. L'application cliente doit disposer d'une paire de clés (privée, publique) et doit pouvoir interpréter les licences ODRL et MPEG-21 REL.

5.2. Communication entre composants

La figure 6 présente notre architecture sous deux parties. Une partie formée du noyau du système DRM. C'est cette partie qui s'occupe de la protection du contenu ainsi que de la délivrance des licences d'utilisation. Dans l'autre portion interviennent les composants externes (`content owner`, `content_distribution`, `users/devices/clients`) dont la présence est nécessaire au bon fonctionnement du système.

L'effectivité de l'interopérabilité repose sur la communication entre le noyau et les entités externes. Aussi, le noyau met-il en place une interface de services web décrite dans un document WSDL pour faciliter les échanges avec les autres entités. Les communications se font alors par échanges de messages SOAP sécurisés [8].

Dans la suite de cette section nous présentons les communications entre éléments d'une paire de composants incluant à chaque fois un composant externe et le composant du noyau qui lui est directement associé. Pour chaque paire nous présentons le squelette des principaux messages qui sont échangés. Le but de ces exemples n'est pas de contraindre la structure des messages qui sont échangés mais de montrer que les informations requises par chaque composant peuvent bien être transmises par la solution proposée, à savoir les messages SOAP sécurisés. Ainsi, nous ne faisons pas apparaître d'appels à méthodes dans les échanges.

5.2.1. Content owner <- -> packager

Le `content owner` chiffre le contenu numérique avec une clé générée de façon aléatoire. La clé générée est chiffrée à son tour en utilisant la clé publique du `packager`. L'ensemble contenu chiffré, clé chiffrée, algorithmes de chiffrement, contrats et droits est alors transmis au `packager` (voir figure 7 (a)). A la réception du message, le `packager` obtient la clé secrète utilisé pour chiffrer le contenu à l'aide de sa clé privée. La ressource protégée est alors déchiffrée à son tour à l'aide de la clé secrète. Si tout se passe bien alors la `packager` retourne au `content owner` un accusé de réception. Sinon un message d'erreur est envoyé (voir figure 7 (b)).

5.2.2. Packager <- -> license server

Le `packager` chiffre le contenu numérique avec une clé générée de façon aléatoire. La clé générée est chiffrée à son tour en utilisant la clé publique du `license server`. Le `packager` marque (watermarking) ensuite le contenu chiffré et envoie l'ensemble contenu protégé, clés chiffrés, algorithmes, contrats et droits au `license server`. La structure du message envoyé est similaire à celle de la figure 7 (a). Le `license server` stocke ces informations dans sa base de données et retourne un accusé similaire à celui de la figure 7 (b) au `packager`.

5.2.3. Packager <- -> content distribution server

L'interface entre le `packager` et le `content_distribution` est une interface de requêtes. Le composant `content_distribution` adresse des requêtes au `packager` par rapport aux contenus qu'il aimerait acquérir. Le `packager` lui retourne les contenus protégés demandés s'ils existent dans sa base.

5.2.4. Content distribution server <- -> users

L'utilisateur télécharge les contenus sécurisés de façon classique. Notons qu'il peut aussi les recevoir sur supports physiques tels que CDs. Il peut aussi les transmettre à d'autres utilisateurs.

5.2.5. Users/devices/clients <- -> license server

Le client DRM et le `license server` communiquent par échanges de messages SOAP sécurisés.

Lorsque le client tente d'exploiter le contenu protégé pour la première fois, il constate que la licence n'est pas disponible. Il extrait l'URL du `license server` du contenu et le contacte pour obtenir une licence. La demande de licence est un message soap contenant l'identifiant du contenu et la clé publique du client (voir figure 8 (a)).

Le `license server` renvoie l'utilisateur vers la composante `AAA_or_payment` pour l'achat d'une licence ou pour authentification. Après authentification et/ou paiement, Le composant `AAA_or_payment` transmet alors au `license server` les éléments définissant le type de licence sollicité par l'utilisateur.

Le `license server` chiffre alors la clé de décryptage du contenu en se servant de la clé publique du client. Sur la base des contrats et des droits liés à ce contenu, il établit une licence (ODRL ou MPEG-21 REL) sur laquelle il appose sa signature. Ensuite il transmet l'ensemble clé chiffrée, signature numérique et licence au client DRM (voir figure 8 (b)).

Le client DRM déchiffre la clé reçue à l'aide de sa clé privée. Après vérification de la signature, il peut alors décrypter la ressource et la rendre à l'utilisateur suivant les directives de la licence.

```
<?xml version="1.0"?>
<soap:envelope>
  <soap:header>
    <wss:security>
      <encryptedKey>
        <encryptionMethod Algorithm="rsa-1_5"/>
        <cipherData>
          <cipherValue>donfdle...</cipherValue>
        </cipherData>
      </encryptedKey>
      <signature> -- optionnel -- </signature>
    </wss:security>
  </soap:header>
  <soap:body>
    <encryptedData>
      <encryptionMethod Algorithm="tripleDES-cbc"/>
      <cipherData>
        <cipherValue>d2dnqye...</cipherValue>
      </cipherData>
    </encryptedData>
    <rights-and-contracts>...</rights-and-contracts>
  </soap:body>
</soap:envelope>
```

(a)

```
<?xml version="1.0"?>
<soap:envelope>
  <soap:header>
    <wss:security>
      <signature>-- optionnel --</signature>
    </wss:security>
  </soap:header>
  <soap:body>
    <ack> accusé de réception </ack>
    <soap:fault>-- optionnel --</soap:fault>
  </soap:body>
</soap:envelope>
```

(b)

Figure 7 – messages échangés entre le content owner et le packager

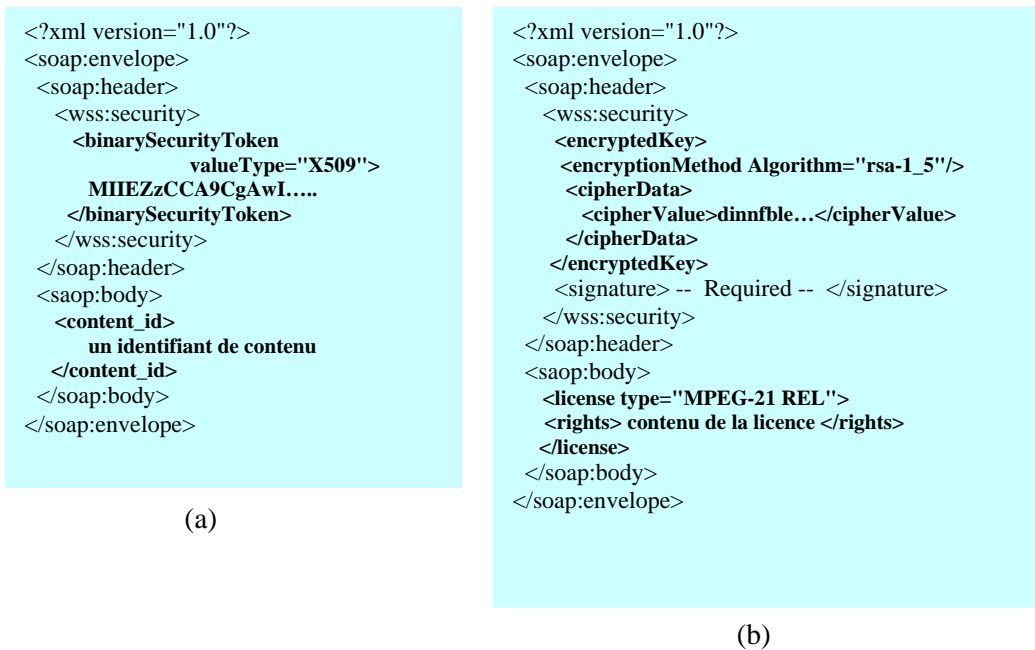


Figure 8 – messages échangés entre clients DRM et le license server

6. Etat de l'art

Dans cette section, nous présentons les travaux existants. Nous comparons à chaque fois notre proposition avec ces travaux.

A notre connaissance, Opera et DMDFusion sont les seuls systèmes existants qui facilitent l'interopérabilité des solutions DRM. DMDFusion [13] incorpore les systèmes Microsoft, Adobe et Real Networks. Il fournit une interface commune pour ces trois solutions. Le contenu est protégé en utilisant les techniques spécifiques à chacune des solutions. Bien qu'intéressante, cette proposition ne prend pas en compte les autres systèmes existants. Opera [12] comble ce manquement en proposant une architecture ouverte pour l'interopérabilité des systèmes DRM. Il s'agit ici aussi d'un interfaçage de solutions existantes. Lorsque le système reçoit une demande de licence, il édite une licence pour le système de DRM d'où provient la requête. Ainsi, en réponse à une requête Real sera éditée une licence Real Network. Nous en déduisons 2 inconvénients majeurs:

- Non seulement toutes les requêtes en vue d'obtenir une licence sont gérées par le client Opera, mais le processus de délivrance de licences est aussi centralisé sur son serveur.
- Si un constructeur change sa technologie ou met sur le marché un nouveau produit, le noyau du système Opera devra être modifié pour prendre en compte les nouvelles données.

Notre proposition n'est pas liée à une solution particulière. Elle utilise les standards existants pour faciliter la communication entre les composants d'une architecture basique de DRM. Notre suggestion est évolutive. En effet il est possible sans modifier les composants présents dans l'architecture d'en ajouter un. Il suffirait de définir l'interface du nouveau composant en prenant en compte l'interface (WSDL) de ceux avec lesquels il échangera (en consultant par exemple un répertoire UDDI). Mieux encore, notre proposition donne la possibilité aux intervenants de l'architecture de définir des interfaces standardisées.

Polo et al. [14] constatent que les langages ODRL et MPEG-21 REL ont des entités différentes mais qui représentent les mêmes informations. En conséquence, ils proposent un mécanisme basé

XSLT [16] (eXtensible Stylesheet Language Transformation) pour la transformation d'une licence exprimée en MPEG-21 REL en une licence équivalente en ODRL et vice versa. Cette idée est ingénieuse, mais manque de sémantique. Elle nécessite donc la définition d'un modèle générique (à l'aide de RDF - Ressource Descriptive Framework [15] par exemple) et le développement d'une ontologie pour ce modèle. Nous proposons dans ce papier l'usage des langages MPEG-21 REL et ODRL comme langages d'expression de droits.

7. Conclusion

Le manque d'interopérabilité entre les différents systèmes DRM a souvent été avancé pour expliquer l'échec des DRM notamment dans le cadre de la distribution de contenu musical. Dans ce papier, nous identifions les obstacles à ce besoin d'interopérabilité et proposons des solutions pour les contourner. Ainsi, nous utilisons les concepts véhiculés par les services web pour définir l'architecture générale d'un système DRM inter opérable. Les différents acteurs de l'architecture communiquent par échanges de messages SOAP.

Nous avons établi qu'il était difficile de faire le choix entre les deux langages d'expression de droits les plus utilisés de la littérature que sont ODRL et MPEG-21 REL. Nous envisageons alors la définition d'un langage générique qui engloberait ODRL et MPEG-21 REL. Autre perspective de ce travail consiste à définir une ontologie des DRM afin de spécifier de façon standard les interfaces des différents acteurs de l'architecture proposée.

8. References

- [1] *Electronic copyright management systems: Requirements, players and technologies*, F. Bartolini, Cappellini, A. Piva, A. Fringuelli, In proceedings of the 10th IEEE International Workshop on Database and Expert Systems Applications, 1999.
- [2] *Security architectures for controlled digital information dissemination*, J. Park, R. Sandhu, J. Schifalacqua, In proceedings of the 16th Annual Computer Security Applications Conference, 2000.
- [3] *Fair use, drm and trusted computing*, J. Erickson, Communications of the ACM Vol. 46, No. 4 (2003), 34-39.
- [4] *ODRL-Open Digital Rights Language*, R. Iannella, W3C Note, <http://www.w3.org/TR/odrl/>, 2002.
- [5] *ISO/IEC 21000:2004 Information technology – Multimedia framework (MPEG-21)*, International Organization for Standardization (ISO), 2004.
- [6] *SOAP - Service Oriented Architecture Protocol*, M. Gudgin, M. Hadley, N. Mendelsohn, J. J. Moreau, H. F. Nielsen, W3C Recommendation, <http://www.w3.org/TR/soap/>, 2003.
- [7] *Namespace in XML 1.0*, T. Bray, D. Hollander, A. Layman, R. Tobin, W3C Recommendation, <http://www.w3.org/TR/REC-xml-names/>, 2006.
- [8] *Web services security: SOAP Message Security 1.1*, OASIS Standard Specification, 2006.
- [9] *XML Signature Syntax and Processing*, M. Bartel, J. Boyer, B. Fox, B. LaMachia, E. Simon, W3C Recommendation, <http://www.w3.org/TR/xmlsig-core/>, 2002.
- [10] *XML Encryption Syntax and Processing*, T. Imamura, B. Dillaway, E. Simon, W3C Recommendation, <http://www.w3.org/TR/xmlenc-core/>, 2002.
- [11] *FORM: a federated rights expression model for open DRM frameworks*, T. Sans, F. Cuppens, N. Cuppens-Boulahia, In proceedings of the 11th Annual Asian Computer Science Conference, 2006.

Interopérabilité des systèmes DRM

- [12] OPERA-Interoperability of Digital Rights Management (DRM) Technologies, S. Wegner, Eurescom Technical Information, <http://www.eurescom.de>, 2003.
- [13] DMD Fusion, DMDSecure/SafeNet, <http://www.safenet-inc.com/>.
- [14] *Interoperability between ODRL and MPEG-21 REL*, J. Polo, J. Prados, J. Delgado, In proceedings of the 1st International ODRL Workshop, 2004.
- [15] *RDF-Ressource Descriptive Framework*, W3C Semantic Web Activity, <http://www.w3.org/RDF/>.
- [16] XSLT, J. Clark, W3C Recommendation, <http://www.w3.org/TR/xslt>, 1999.