Gestion sémantique des droits d'accès au contenu: l'ontologie AMO

Michel Buffa, Catherine Faron Zucker, Anna Kolomoyskaya

Gestion sémantique des droits d'accès au contenu: l'ontologie AMO

Michel Buffa*, Catherine Faron-Zucker*
Anna Kolomoyskaya*

*I3S, Université de Nice Sophia Antipolis, CNRS
930 route des Colles - BP 145
FR-06903 Sophia Antipolis cedex
{michel.buffa,catherine.faron-zucker}@unice.fr

Résumé. Dans cet article nous proposons une approche de la gestion des droits d'accès pour les systèmes de gestion de contenu qui reposent sur les modèles et techniques du web sémantique. Nous présentons l'ontologie AMO qui consiste (1) en un ensemble de classes et propriétés permettant d'annoter les ressources dont il s'agit de contrôler l'accès et (2) en une base de règles d'inférence modélisant la stratégie de gestion des droits à mettre en œuvre. Appliquées sur la base d'annotations des ressources, ces règles permettent de gérer les ressources selon une stratégie donnée. Cette modélisation garantit ainsi l'adaptabilité de l'ontologie à différentes stratégies de gestion des droits d'accès. Nous illustrons l'utilisation de l'ontologie AMO sur les documents du projet ANR ISICIL produits par le wiki sémantique SweetWiki. Nous montrons comment les documents sont annotés avec AMO, quelles règles sont mises en œuvre et quelles requêtes permettent le contrôle de l'accès aux documents.

1 Introduction

La sécurité, la protection, le contrôle de l'accès représentent un enjeu majeur des systèmes de gestion de contenu. Cette problématique est centrale dans les sites web collaboratifs et dans les réseaux sociaux du web 2.0 où le principe même de l'édition collaborative de documents et la notion de partage posent la question de la définition des droits d'accès. De fait, le contrôle de l'accès aux ressources est un des défis lancés au web sémantique.

Dans cet article nous nous intéressons aux systèmes de gestion de contenu qui reposent sur des serveurs web sémantiques et nous proposons une approche de la gestion des droits d'accès aux ressources fondée sur les modèles et techniques du web sémantique. Plus précisément, nous présentons une ontologie dédiée à la représentation des droits d'accès donnés sur un document à certains utilisateurs ou classes d'utilisateurs. Nous avons baptisé cette ontologie AMO, acronyme signifiant *Access Management Ontology*. AMO consiste d'une part en un ensemble de classes et propriétés permettant d'annoter les ressources dont il s'agit de gérer l'accès et en une base de règles d'inférence modélisant la stratégie (*policy* en anglais) de contrôle des droits à mettre en œuvre. Appliquées sur la base d'annotations des ressources,

ces règles permettent de contrôler l'accès à ces ressources selon une stratégie donnée. Cette modélisation déclarative sous forme d'une base de règles assure une adaptation facile de l'ontologie à différentes stratégies de contrôle des droits d'accès qui dispense ainsi de modifier les annotations des documents dans le cas d'un changement de stratégie.

Dans le cadre du projet ANR ISICIL [1], nous utilisons l'ontologie AMO pour gérer l'accès aux ressources partagées par un réseau de veilleurs d'entreprise : documents produits par des outils de gestion de contenu, wikis ou blogs, documents HTML statiques, dont certains issus d'opérations de web-scraping, bookmarks, etc. Un des enjeux de ce projet orienté autour des techniques du web 2.0 et du web sémantique concerne la gestion de l'accès aux ressources partagées par le réseau social des veilleurs d'entreprise. Parmi les documents produits par les veilleurs figurent ceux d'un site collaboratif géré par le wiki sémantique SweetWiki que nous développons (Buffa et al., 2008) et qui nous servira à illustrer dans cet article l'utilisation d'AMO. SweetWiki intègre les technologies du web sémantique pour améliorer la structure, la recherche et la navigation. Plus précisément, il associe aux pages wiki des annotations RDF/S qui rendent le contenu de ces pages exploitable par le moteur sémantique CORESE (Corby et al., 2004).

Nous présentons l'ontologie AMO dans la partie 2 puis nous montrons dans la partie 3 l'utilisation qui est faite d'AMO dans SweetWiki et ce faisant nous mettons en lumière la facilité d'adaptation d'AMO à différentes stratégies de contrôle. La section 4 est dédiée au positionnement de notre approche par rapport aux travaux existants sur la gestion de l'accès aux ressources dans les systèmes de gestion de contenu et par rapport aux modèles sémantiques du web 2.0.

2 L'ontologie AMO

Dans un système de fichiers comme dans un système de gestion de contenu, des rôles (administrateur, propriétaire, etc.) sont associés aux utilisateurs ou catégories d'utilisateurs et différents types d'accès aux ressources (écriture, lecture, etc.) sont définis, l'accès aux ressources variant d'un utilisateur à l'autre selon son rôle. Cette analyse nous a conduits à définir un ensemble de classes et propriétés nécessaires à la description des droits d'accès à une ressource. C'est ce que nous décrivons dans la partie 2.1.

Si les systèmes de gestion de contenu partagent les mêmes principes généraux pour le contrôle de l'accès aux ressources, ils adoptent cependant des stratégies qui peuvent varier d'un système à l'autre. Pour permettre une adaptation facile de l'ontologie sur laquelle reposera la gestion de l'accès aux ressources en fonction de la stratégie adoptée, nous modélisons dans AMO cette stratégie sous la forme déclarative d'une base de règles d'inférence qui peuvent être modifiées à loisir sans incidence sur la base d'annotations des ressources à gérer. Nous décrivons dans la partie 2.2 une base de règles correspondant à la stratégie de contrôle de l'accès aux ressources adoptée dans le wiki sémantique SweetWiki.

1. http ://isicil.inria.fr/

2.1 Classes et propriétés de AMO

AMO repose sur quelques principes fondateurs que partagent tous les systèmes de gestion de contenu :

- Les *agents* d'un système de gestion de contenu sont les utilisateurs, groupes d'utilisateurs, services qui interagissent avec le système.
- Ces agents ont des *rôles*. Dans le cas de systèmes d'édition collaborative tels que les wikis ou les CMS, il s'agit des rôles d'invité (agent non enregistré dans le système), de contributeur, d'administrateur. D'autres rôles peuvent être modélisés selon le type de systèmes.
- A chaque rôle est associée une liste d'*actions* autorisées. Dans le cas des systèmes d'édition collaborative, les actions possibles sur une ressource sont la création, la lecture, la modification et la destruction de contenu, la modification des droits, la modification de la liste des agents autorisés sur une ressource, la modification du type d'accès défini pour une ressource. D'autres actions peuvent être modélisées pour d'autres types de systèmes.
- Il existe différents *types d'accès* aux ressources du système. Dans le cas des systèmes d'édition collaborative, une ressource peut être publique (tous les utilisateurs y ont accès en lecture et écriture), privée (seuls les agents autorisés ont accès en lecture et écriture) ou semi-privée (accès libre en lecture, accès en écriture uniquement pour les agents autorisés). Là encore, d'autres types d'accès peuvent être ajoutés pour d'autres types de systèmes.
- Enfin, les actions autorisées à un agent sur une ressource dépendent du rôle de l'agent et/ou du type d'accès défini pour la ressource.

L'ontologie AMO présentée sur la figure 1 regroupe les concepts nécessaires pour représenter ces connaissances. Les trois classes `Role`, `Action` et `AccessType` sont centrales dans AMO. `Role` est la méta-classe des classes `Administrator`, `Contributor` et `Guest`. `Action` est la méta-classe des classes `ReadContent`, `ModifyContent`, `DeleteContent`, `ModifyUserRights`, `ModifyAccessType` et `ModifyAuthorizedAgents`. Enfin, `AccessType` est la méta-classe des classes `Private`, `Public` et `SemiPublic`.

Trois classes du vocabulaire FOAF - standard du web social présenté dans la section 4 - sont également centrales dans AMO : la classe `Agent` et sa sous-classe `Group`, et la classe `Document`. Elles sont utilisées comme domaine ou codomaine de propriétés AMO ainsi que dans les règles AMO.

Les propriétés `creator` et `hasAuthorizedAgent` associent un agent à un document (elles ont pour domaine la classe `Document` et pour codomaine la classe `Agent`) ; la propriété `hasRole` associe un rôle à un agent et la propriété `hasActionOnResource` associe une action à un rôle ; la propriété `hasAccessType` associe un type d'accès à un document.

En outre, pour représenter dans un modèle de propriétés binaires la relation ternaire qui stipule qu'un agent est autorisé à effectuer une action sur une ressource, nous avons réifié cette relation en introduisant une sous-classe `AuthorizedActionOnResource` de la classe `Action`, une propriété `hasAuthorizedActionOnResource` qui associe une instance de `AuthorizedActionOnResource` à un agent, et les propriétés `hasDocument` et `hasAction` qui associent à une instance de `AuthorizedActionOnResource` respectivement un document et une action.

AMO est un vocabulaire RDFS, qui peut donc être utilisée pour annoter dans le langage RDF les ressources dont il s'agit de contrôler l'accès.

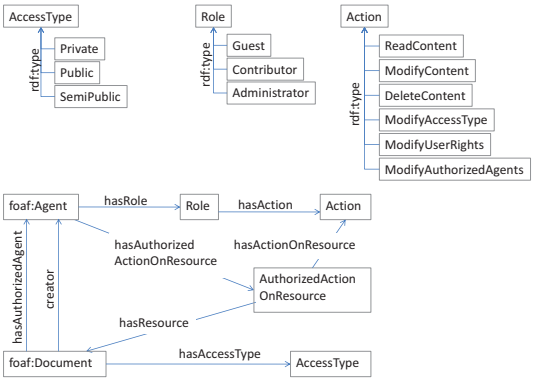

FIG. 1 – *Classes et propriétés de l'ontologie AMO.*

2.2 Base de règles de AMO

Les systèmes de gestion de contenu adoptent des stratégies de contrôle d'accès aux ressources qui peuvent varier d'un système à l'autre. Plutôt que de faire varier les annotations des ressources en fonction des stratégies de contrôle auxquelles ces ressources seront soumises, nous proposons de modéliser de façon déclarative la stratégie de contrôle dans l'ontologie AMO, sous la forme d'une base de règles, certaines règles pouvant varier en fonction de la stratégie modélisée. La base de règles présentée ici est celle de SweetWiki dont la stratégie de contrôle d'accès s'inspire de celle du moteur de wiki open source DekiWiki [2] très largement utilisé.

Par défaut, les administrateurs ont tous les droits sur toutes les ressources. Les contributeurs ont tous les droits relatifs au contenu des ressources ; ceux déclarés agents d'une ressource par l'auteur de celle-ci ont en outre certains droits d'administration sur cette ressource. Les invités ont uniquement le droit de lecture du contenu des ressources. La figure 2 ci-après résume les droits d'accès à une ressource en fonction de son type d'accès et du rôle de l'utilisateur qui cherche à accéder à la ressource (horizontalement figurent les types d'accès des ressources, verticalement les rôles utilisateurs).

Nous modélisons cette stratégie dans AMO de façon *déclarative* par six règles d'inférence correspondant chacune à une situation décrite sur la figure 2. Par exemple, la règle 1 ci-après spécifie les droits attribués aux agents donnés à une ressource. D'autres règles stipulent des lois générales telles que *un membre d'un groupe hérite du ou des rôle(s) attribués à son groupe* (règle 2) ou encore *le créateur d'une ressource est un agent de cette ressource* (règle 3).

Ces règles sont exprimées dans le langage SPARQL en utilisant des requêtes de la forme CONSTRUCT / WHERE : une telle requête permet de *construire* des graphes RDF en substituant aux variables de sa clause CONSTRUCT les valeurs qui satisfont sa clause WHERE (retrouvées en recherchant les appariements possibles de sa clause WHERE avec les données RDF inter-

2. http ://www.mindtouch.com/

	Public	Semi-Public	Private
Guest	ReadContent	ReadContent	
Contributor	ReadContent	ReadContent ModifyContent DeleteContent	
AuthorizedAgent	ReadContent ModifyContent DeleteContent ModifyAuthorizedAgents ModifyAccessType		
Administrator	ReadContent ModifyContent DeleteContent ModifyAuthorizedAgents ModifyAccessType ModifyUserRights		

FIG. 2 – *Une stratégie de contrôle des droits d'accès modélisée dans AMO.*

rogées). Une requête CONSTRUCT/WHERE peut donc être vue comme une règle appliquée en chaînage avant la clause WHERE étant la prémisse et la clause CONSTRUCT la conclusion. Ce format de règle peut cependant aussi bien être utilisé pour des règles appliquées en chaînage arrière, comme c'est le cas dans le moteur sémantique Corese.

Règle 1

```
CONSTRUCT {
   ?agent amo:hasAuthorizedActionOnResource ?a
   ?a amo:hasResource ?resource
   ?a amo:hasActionOnResource amo:ReadContent.
   ?a amo:hasActionOnResource amo:ModifyContent.
   ?a amo:hasActionOnResource amo:DeleteContent.
   ?a amo:hasActionOnResource amo:ModifyAccessType.
   ?a amo:hasActionOnResource amo:ModifyAuthorizedAgents }
WHERE {
   ?resource rdf:type foaf:Document.
   ?resource amo:hasAuthorizedAgent ?agent }
```

Règle 2

```
CONSTRUCT {
  ?agent amo:hasRole ?role }
WHERE {
  ?group amo:hasRole ?role
  ?group foaf:member ?agent }
```

Règle 3

```
CONSTRUCT ?resource amo:hasAuthorizedAgent ?agent
WHERE ?resource amo:creator ?agent
```

Cette modélisation *déclarative* de la stratégie de gestion des droits d'accès assure une grande facilité de maintenance. Le changement des droits d'une catégorie d'utilisateurs - et ce pour toutes les ressources concernées - ne nécessitera que l'ajout ou la suppression de triplets dans la conclusion d'une règle. De même, l'ajout de nouveaux rôles ne demanderait que l'ajout d'une classe représentant ce rôle et des règles représentant les droits d'accès associés à ce rôle.

3 Gestion des droits d'accès dans SweetWiki

L'ontologie AMO est utilisée dans le projet ISICIL pour annoter des ressources partagées par un réseau social de veilleurs d'entreprise. La gestion de l'accès à ces ressources dans le moteur SweetWiki repose sur (1) l'exploitation de ces annotations sémantiques, (2) des inférences sur ces annotations basées sur les règles d'AMO et (3) la formulation de requêtes SPARQL afin de retrouver des connaissances relatives aux accès autorisés à un utilisateur donné sur une ressource donnée. Dans SweetWiki les annotations des ressources reposent sur les ontologies FOAF, SIOC et AMO et des requêtes SPARQL sont utilisées dans la plupart des fonctionnalités implémentées : les annotations RDF nourrissent le moteur sémantique CORESE embarqué dans SweetWiki. En particulier, en utilisant les possibilités de recherche approchée de Corese(Corby et al., 2006) et un système de tagging sémantique des documents, SweetWiki offre une navigation "intelligente" au travers de suggestions.

3.1 Annotation des ressources d'ISICIL avec AMO

Lors de la création d'une page wiki, l'identité de son créateur est enregistrée ainsi que le type d'accès à la page décidé par celui-ci et éventuellement un ou plusieurs agents autorisés sur la page, désignés également par le créateur. Dans SweetWiki ces informations sont traduites sous la forme d'annotations RDF associées aux pages créées. Par exemple, l'annotation 1 suivante résulte de la création d'une page wiki privée par l'utilisateur `AnnaKolomoiska` qui déclare `MichelBuffa` agent autorisé de cette page. Cette annotation utilise les propriétés `creator`, `hasAuthorizedAgent` et `hasAccessType` d'AMO (ainsi que la classe `WikiArticle` de SIOC - standard du web social présenté dans la section 4).

Annotation 1

```
<rdf:RDF xmlns="http://seetwiki.i3s.unice.fr/AMO.rdfs#" ... >
<sioc:WikiArticle rdf:about="#TestPage">
...
    <creator rdf:resource="#AnnaKolomoiska"/>
    <hasAuthorizedAgent rdf:resource="#MichelBuffa"/>
    <hasAccessType rdf:resource="#Private"/>
  </sioc:WikiArticle>
</rdf:RDF>
```

Lors de l'enregistrement d'un utilisateur sur SweetWiki, cette information est également représentée sous la forme d'une annotation RDF. Par exemple, l'annotation 2 suivante stipule que `MichelBuffa` est un contributeur du wiki. Elle utilise pour cela la classe `Contributor` et la propriété `hasRole` d'AMO (et la classe `Agent` de FOAF).

Annotation 2

```
<rdf:RDF xmlns="http://seetwiki.i3s.unice.fr/AMO.rdfs#" ... >
  <foaf:Agent rdf:about="#MichelBuffa">
   ...
   <hasRole rdf:resource="#Contributor"/>
  </foaf:Agent>
</rdf:RDF>
```

D'autres annotations expriment des connaissances relatives aux groupes d'utilisateurs du wiki. Par exemple, l'annotation 3 stipule que `AnnaKolomoiska` et `CatherineFaron` sont membres du groupe des administrateurs du wiki. Elle utilise pour cela la propriété `hasRole` d'AMO (et les classes `Group` et `Agent` et la propriété `member` de FOAF).

Annotation 3

```
<rdf:RDF xmlns="http://seetwiki.i3s.unice.fr/AMO.rdfs#" ... >
 <foaf:Group rdf:about="#AdminGroup">
  <foaf:member>
   <foaf:Agent rdf:about="#AnnaKolomoiska"/>
  </foaf:member>
  <foaf:member>
   <foaf:Agent rdf:about="#CatherineFaron"/>
  </foaf:member>
  <hasRole rdf:resource="#Admin"/>
 </foaf:Group>
</rdf:RDF>
```

3.2 Inférences exploitant la base de règles de AMO

Appliquées sur les annotations des ressources d'ISICIL, les règles de l'ontologie AMO permettent d'inférer les droits des utilisateurs du wiki sur ces ressources. Considérons par exemple à nouveau la règle 1 qui illustre la section 2.2. Sa prémisse s'apparie avec l'annotation 1 qui illustre la partie 3.1 : la ressource `TestPage` est de type `WikiArticle` - une classe du vocabulaire SIOC, sous-classe de la classe `Document` du vocabulaire FOAF, et `TestPage` est en relation `hasAuthorizedAgent` avec l'utilisateur `MichelBuffa`. Appliquée sur l'annotation 1, la règle 1 permet de conclure que l'utilisateur `MichelBuffa` a les droits de lecture, modification et destruction du contenu de la ressource `TestPage` annotée et ceux de modification de son type d'accès et de sa liste d'agents.

De la même manière, la règle 2 appliquée sur l'annotation 3 permet de déduire que l'utilisateur `CatherineFaron` possède le rôle d'administrateur. Une autre règle d'AMO décrivant les droits d'un agent ayant le rôle d'administrateur sur toute ressource permet alors de conclure que l'utilisateur `CatherineFaron` possède tous les droits sur la ressource `TestPage`.

Enfin, les règles 1 et 3 appliquées sur l'annotation 1 permettent de déduire que l'utilisateur `AnnaKolomoiska` créateur de la ressource `TestPage` a les droits d'un agent sur cette ressource (lecture, modification et destruction du contenu et modification du type d'accès et de la liste d'agents).

3.3 Requêtes SPARQL pour la gestion des droits d'accès

L'accès à une ressource particulière par un utilisateur donné dépend, comme l'ensemble des actions dans SweetWiki, de la réponse du moteur CORESE à une requête sémantique lancée sur la base d'annotations des ressources. Pour cela, CORESE combine chaînage arrière sur la base de règles AMO et appariement de graphes requêtes sur la base d'annotations. Par exemple, la requête SPARQL suivante indique si l'utilisateur `CatherineFaron` possède le droit de modification du contenu de la ressource `TestPage` :

Requête 1

```
prefix amo: <http://sweetwiki.unice.fr/AMO.rdfs#>
ASK {
 <http://sweetwiki.unice.fr#CatherineFaron>
                    amo:hasAuthorizedAccessOnResource ?x
 ?x amo:hasActionOnResource amo:ModifyContent
 ?x amo:hasResource <http://sweetwiki.unice.fr#TestPage> }
```

En fonction des différentes fonctionnalités de SweetWiki, d'autres requêtes sont mises en œuvre. Par exemple, la requête SPARQL suivante permet de retrouver la liste des utilisateurs ayant des droits sur la ressource `TestPage` et pour chacun la liste de ses actions autorisées sur `TestPage` :

Requête 2

```
prefix amo: <http://sweetwiki.unice.fr/AMO.rdfs#>
SELECT ?agent ?action {
 ?agent amo:hasAuthorizedAccessOnResource ?x
 ?x amo:hasActionOnResource ?action
 ?x amo:hasResource <http://sweetwiki.unice.fr#TestPage> }
order by ?agent
```

4 Positionnement

4.1 Les langages XML pour le contrôle d'accès et les droits numériques

La plupart des mécanismes de contrôle d'accès implémentés dans les systèmes de gestion de contenus sont basés sur des langages XML permettant de décrire des politiques de contrôle d'accès et de gestion des droits numériques (ou DRM, acronyme pour Digital Rights Management). Ces systèmes exploitent des métadonnées associées aux ressources dont il s'agit de contrôler l'accès et ces métadonnées respectent les schémas XML correspondant à ces langages dédiés. Parmi ceux-ci, les plus connus sont XrML [3] (eXtensible Right Markup Language) utilisé comme base du langage d'expression des droits du standard MPEG-21 [4], ODRL [5](Open

3. http ://www.xrml.org/
4. http ://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
5. http ://www.w3.org/TR/odrl/

Digital Right Language) implémenté par l'Open Mobile Alliance (OMA) et XACML [6] (Extensible Access Control Markup Language) développé par OASIS. Le modèle ODRL repose sur les notions d'*Asset*, *Party*, *Permission*, *Constraint*, *Requirement*, *Condition*, *Rights holder*, *Context*, *Offer*, *Agreement*, et *Revoking rights*. Le modèle XACML permet de représenter des stratégies de contrôle d'accès sous forme de règles et repose sur les notions de *Rule*, *Policy* et *Policy Set*, ces notions pouvant être précisées avec celles de *Subject*, *Resource*, *Action*, *Environment*. Une *Rule* comprend des *Conditions* et *Effects*, et une *Policy* des *Rules* et *Obligations*.

4.2 Les approches sémantiques du contrôle d'accès

Avec l'émergence du web 2.0 et du web sémantique, de nouveaux systèmes de gestion de contenu ont vu le jour qui reposent sur ces nouvelles approches du web pour gérer l'accès au contenu. Notamment, (Alam et al., 2006) montrent bien les limites des solutions de gestion des droits d'accès utilisant des langages de description non sémantiques. Ils proposent une ontologie OWL pour décrire les accès à des services web, inspirée du modèle XACML. Plus généralement, nous retrouvons dans les quelques modèles sémantiques de gestion de l'accès au contenu qui existent certaines notions déjà présentes dans les langages XML plus anciens.

Citons également le système de contrôle d'accès basé sur RDF qu'utilise le W3C depuis 2001 pour contrôler l'accès aux fichiers de ses serveurs : W3C ACL System [7]. Récemment, (Hollenbach et al., 2009) ont proposé une évolution de ce système pour contrôler les accès de manière décentralisée, à grande échelle, la partie autorisation du système associant à un répertoire une liste d'accès sous forme de métadonnées RDF. L'ontologie utilisée se nomme Basic Access Control Ontology [8] et est présentée comme une base solide pour développer des systèmes plus sophistiqués. Contrairement à l'ontologie AMO présentée dans cet article, le contrôle d'accès à base de règles d'inférence n'est pas proposé (bien que les auteurs l'évoquent comme une possible future évolution), ni le contrôle au niveau des documents (plutôt qu'à celui des répertoires).

Les besoins en termes de gestion des droits d'accès dans le projet ISICIL s'apparentent à ceux des bibliothèques numériques dont Coyle (2004) propose une synthèse. Cependant, une des problématiques essentielles aux bibliothèques numériques n'est pas pertinente dans le cadre d'ISICIL : celle du respect des copyrights des documents rendus accessibles et pour ce faire la protection des documents par des DRMs. Les documents manipulés par les veilleurs demeurent en effet dans l'intranet des entreprises ou bien sont des documents publics sur le web. Parmi les travaux sur la gestion des accès dans les bibliothèques numériques, citons ceux de (Lagoze et al., 2006) sur l'architecture Fedora de gestion des ressources numériques et ceux de Kruk et al. (2008) sur la bibliothèque numérique sémantique JeromeDL.

Les auteurs de Fedora proposent un modèle appelé DARS (acronyme de Distributed Active Relationships) permettant d'associer des métadonnées aux objets d'une bibliothèque numérique, en particulier pour la gestion des droits d'accès. Bien qu'une partie du modèle de gestion des accès se trouve ainsi dans une ontologie, le système Fedora utilise cependant également des métadonnées XACML associées aux ressources qu'il manipule.

6. http ://www.oasis-open.org/committees/xacml/
7. http ://www.w3.org/2001/04/20-ACLs
8. http ://www.w3.org/ns/auth/acl

La gestion des accès dans JeromeDL repose sur l'ontologie EAC [9] (acronyme pour Extensible Access Control) développée par Kruk (2008). EAC permet d'associer des licences à des ressources, chaque licence correspondant à une politique d'accès. Par exemple, une licence peut indiquer que seules les personnes d'une organisation donnée peuvent accéder aux ressources associées. Le but d'EAC est donc de *filtrer* l'accès aux ressources alors que celui d'AMO est de définir des droits d'accès associés à des rôles d'utilisateur.

4.3 Les standards du web social et du web de données

Une spécificité de l'approche que nous adoptons avec AMO est de s'intégrer aux modèles adoptés dans les domaines du web social et du web de données. Plus précisément, SweetWiki utilise les ontologies FOAF et SIOC pour annoter ses ressources et AMO vient compléter ces ontologies pour permettre de gérer l'accès au contenu. FOAF [10] (acronyme de Friend Of A Friend) est un vocabulaire RDF utilisé dans les réseaux sociaux pour décrire les personnes et les liens qu'elles entretiennent entre elles. SIOC [11] (acronyme de Semantically-Interlinked Online Communities) est un autre vocabulaire RDF qui modélise les concepts des applications sociales du web : forums, les blogs, les wikis. Il réutilise certains concepts de FOAF et d'autres ontologies populaires (Dublin Core, SKOS, etc.) et s'est imposé comme standard. Il est aujourd'hui intégré dans de nombreuses applications comme le moteur de blog WordPress et son adoption au sein du projet *Linked Data* [12] confirme sa popularité. Un module de SIOC est prévu depuis longtemps pour la modélisation des droits d'accès [13] mais est resté vide jusqu'à présent. Tout récemment, en fin de rédaction de cet article (20 septembre 2009), ce module a été complété. Nous prévoyons à court terme d'étudier l'alignement des deux ontologies. Probablement, les classes `Role`, `Action` et `AccessType` d'AMO pourront être alignées avec celles respectivement de `Role` dans *SIOC Core Ontology*, et `Permission` et `Status` nouvellement ajoutées dans le module *SIOC Access Ontology*.

FOAFRealm est une extension de FOAF proposée pour filtrer collaborativement l'accès aux ressources en fonction des profils des utilisateurs et des relations qu'ils entretiennent dans un réseau social. Ce vocabulaire est par exemple utilisé dans JeromeDL pour un filtrage basé sur des mesures de confiance dans un réseau social. Un tel filtrage peut être complémentaire du contrôle d'accès basé sur les rôles des utilisateurs et les types d'accès aux ressources que permet AMO.

Enfin, le problème de l'autorisation d'accès aux ressources auquel répond AMO est connexe à celui de l'authentification des agents qu'il s'agira d'aborder dans le cadre général du projet ISICIL. Nous envisageons pour cela de reposer sur le protocole FOAF-SSL(Story et al., 2009).

5 Conclusion et Perspectives

Nous avons présenté dans cet article l'ontologie AMO dédiée à la gestion de l'accès dans les systèmes de gestion de contenu. AMO consiste d'une part en un ensemble de classes et pro-

9. http ://www.jeromedl.org/eac/1.0/spec/index.html
10. http ://xmlns.com/foaf/spec/
11. http ://sioc-project.org/ontology
12. http ://linkeddata.org/
13. http ://rdfs.org/sioc/access

priétés qui permettent d'annoter les ressources dont il s'agit de contrôler l'accès. AMO propose d'autre part une base de règles d'inférence qui représentent de façon déclarative une stratégie d'accès aux ressources ; cette base peut être facilement modifiée, sans incidence sur les annotations des ressources, en fonction de la stratégie à mettre en œuvre sur telle ou telle application web à gérer. Nous montrons l'utilisation qui peut être faite d'AMO par des exemples d'annotations de ressources dans SweetWiki et nous validons ce premier prototype par des requêtes SPARQL permettant d'interroger la base d'annotations relatives aux accès à l'aide du moteur sémantique CORESE. AMO s'appuie sur FOAF et sur SIOC et se rend ainsi compatible avec les applications qui ont déjà basé leur développement sur ces deux ontologies populaires pour décrire leurs ressources. AMO n'utilise cependant pas les concepts du module `SIOC access` dédié au contrôle d'accès qui vient d'être publié sur le web il y a quelques jours à peine ; un travail d'alignement ontologique est maintenant à faire.

L'utilisation d'une base de règles permet de s'affranchir de l'implémentation complexe des mécanismes de calcul des droits, qui font appel à l'héritage (héritage des rôles des groupes auquel appartient un agent, union des actions permises par ces rôles), à l'ordre des opérateurs (interdire puis autoriser, ou le contraire...), à des mécanismes implicites (le créateur d'une ressource bénéficie automatiquement de certains droits sur cette même ressource). Dans l'état de l'art nous avons vu qu'aucune des approches sémantiques de gestion des accès n'utilise de règles, les auteurs de l'ontologie ACL du W3C ayant simplement évoqué cette possibilité d'extension. Comparée aux langages XML antérieurs au web sémantique, comme XACML qui bénéficient de bibliothèques logicielles importantes et complexes pour implémenter toutes les stratégies de gestion des droits, et qui ont demandé un travail de développement considérable, AMO en revanche demeure simple à mettre en œuvre et à étendre (par exemple en réutilisant l'ontologie FOAF Realm pour définir des règles d'accès basées sur des mesures de confiance).

Nous avons présenté des exemples d'utilisation d'AMO avec les documents produits par SweetWiki. AMO est également utilisée par d'autres applications dans le cadre du projet ANR ISICIL. Précisément, les classes et propriétés d'AMO sont utilisées dans les annotations des documents du serveur des profils des utilisateurs. L'ontologie SemSNI utilisée pour ces annotations, sert à modéliser les interactions entre utilisateurs d'un réseau social. Elle fait référence à AMO pour la définition des droits d'accès aux ressources partagées par chaque utilisateur. SemSNI et AMO ont été également utilisées pour spécifier les accès aux documents partagés par les utilisateurs du réseau social Ipernity.com (un site français inspiré de FaceBook spécialisé sur le partage de photos) Erétéo et al. (2009). Prototypée dans SweetWiki, la couche de gestion des accès basée sur AMO est en cours d'implémentations dans les différentes applications du projet ISICIL.

Remerciements

Ce travail s'inscrit dans le cadre du projet ISICIL financé par l'ANR.

Références

Alam, A., G. Subbiah, B. Thuraisingham, et L. Khan (2006). Reasoning with Semantics-aware Access Control Policies for Geospatial Web Services. In *3rd ACM Workshop On Secure Web*

Services, SWS 2006, pp. 69–76.

Buffa, M., F. Gandon, G. Erétéo, P. Sander, et C. Faron (2008). Sweetwiki : A semantic wiki. *Journal of Web Semantics 6*(1), 84–97.

Corby, O., R. Dieng-Kuntz, et C. Faron-Zucker (2004). Querying the semantic web with corese search engine. In *16th Eureopean Conference on Artificial Intelligence, ECAI 2004*, pp. 705–709. IOS Press.

Corby, O., R. Dieng-Kuntz, C. Faron-Zucker, et F. Gandon (2006). Searching the Semantic Web : Approximate Query Processing Based on Ontologies. *IEEE Intelligent Systems 21*(1), 20–27.

Coyle, K. (2004). Rights Management and Digital Library Requirements. *Ariadne 40*.

Erétéo, G., M. Buffa, F. Gandon, et O. Corby (2009). Analysis of a Real Online Social Network using Semantic Web Frameworks. In *8th International Semantic Web Conference, ISWC 2009*, LNCS. Springer.

Hollenbach, J., J. Presbrey, et T. Berners-Lee (2009). Using RDF Metadata to Enable Access Control on Social Semantic Web. In *International Semantic Web Conference, ISWC 2009*, LNCS. Springer.

Kruk, S. R. (2008). *Extensible Access Control (EAC) Ontology Specification*. DERI, http://www.jeromedl.org/eac/1.0/spec/index.html/.

Kruk, S. R., M. Cygan, et A. Gzella (2008). JeromeDL - Semantic and Social Technologies for Improving User Experience in Digital Libraries. In *World Wide Web Conference, WWW 2008*. ACM.

Lagoze, C., S. Payette, E. Shin, et C. Wilper (2006). Fedora: an Architecture for Complex Objects and their Relationships. *Int. J. on Digital Libraries 6*(2), 124–138.

Story, H., B. Harbulot, I. Jacobi, et M. Jones (2009). FOAF+SSL: RESTful Authentication for the Social Web. In *ESWC Workshop Trust and Privacy on the Social and Semantic Web, SPOT 2009*.

Summary

In this paper we propose an approach to manage access in content management systems which relies on semantic web models and technologies. We present the AMO ontology which consists (1) in a set of classes and properties dedicated to the annotation of the resources whose access should be controlled and (2) in a base of inference rules modeling the access management strategy to carry out. When applied to the annotations of the resources whose access should be controlled, these rules enable to manage their access according to a given strategy. This modelisation thus ensures the adaptability of the AMO ontology to any access management strategy. We illustrate the use of AMO on the documents of a collaborative web-site managed by the semantic wiki SweetWiki in the ANR ISICIL project. We show how to annotate documents with AMO, which AMO inference rules are applied and which semantic queries finally enable to control access to SweetWiki documents.