Ontology-Based Access Rights Management

Michel Buffa, Catherine Faron Zucker

Ontology-Based Access Rights Management

Michel Buffa and Catherine Faron-Zucker

Abstract In this paper we propose an approach to manage access rights in a 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 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 access according to a given strategy. This modelisation is flexible, extendable and ensures the adaptability of the AMO ontology to any access management strategy. We illustrate the use of AMO on the documents of a collaborative website managed by the semantic wiki SweetWiki in the ANR ISICIL project. We show how to annotate documents with AMO, we explain which AMO inference rules can be applied and which semantic queries finally enable to control access to SweetWiki documents.

1 Introduction

Security, protection, access control represent a major challenge in content management systems. This issue is central to collaborative Web sites and social networks of Web 2.0 where collaborative editing of documents and sharing raises the question of the definition of access rights. Management of access to resources is thus one of the challenges facing the semantic web.

In this paper we consider content management systems based on semantic web servers and we propose an approach for managing access rights to resources based on semantic web models and techniques. We present an ontology dedicated to the representation of the access rights given on a document to some users or user classes.

I3S, Université de Nice Sophia Antipolis, CNRS
930 route des Colles - BP 145, FR-06903 Sophia Antipolis cedex, France,
e-mail: michel.buffa@unice.fr, catherine.faron-zucker@unice.fr

We call this ontology AMO, an acronym meaning *Access Management Ontology*. AMO is made of a set of classes and properties for annotating the resources and a base of inference rules modeling the access control policy. When applied to the annotations of resources, these rules enable to control access according to a given strategy. This declarative modeling into a rule base ensures an easy adaptation of the ontology to different access control policies and thus avoids modifying annotations of documents in the case of a change of strategy.

In the framework of the ANR project ISICIL[1], we use the AMO ontology to manage access to resources shared by a network of technical watchers: documents produced by content management tools, wikis or blogs, static HTML documents produced by web scraping (i.e. firefox extensions similar to "'Scrapbook"'), bookmarks, etc.. One of the issues of this project oriented to Web 2.0 and Semantic Web techniques concerns the management of access to the resources shared by the social network of watchers.

Among the documents produced by the watchers are those of a collaborative website run by the semantic wiki SweetWiki that we develop [1] and that will be used in this paper to illustrate the use of AMO. SweetWiki integrates semantic web technologies to improve structure, search and navigation. More specifically, it associates to wiki pages RDF/S annotations that make the content of these pages processable by the semantic engine Corese [2].

We present the ontology AMO in Section 2. Then we show in Section 3 the use of AMO in SweetWiki and in so doing we highlight the adaptability of AMO to different control policies. Section 4 is dedicated to the positioning of our approach compared to existing work on managing access to resources in content management systems and to semantic models of web 2.0.

2 Ontologie AMO

In a file system or in a content management system, roles (administrator, owner, etc.) are associated with users or user groups and different types of access to resources (writing, reading, etc..) are defined, access to resources varying from one user to another depending on its role. This analysis led us to define a set of classes and properties to describe the access rights to resources. This is what we describe in Section 2.1.

Content management systems share the same general principles for access control to resources, however they adopt strategies that may vary from one system to another. To allow easy adaptation of the ontology supporting the management of access to resources according to the chosen strategy, this latter is declaratively modeled in AMO as a base of inference rules that can be modified at leisure without affecting the annotations of the resources to manage. We describe in Section 2.2

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

a rule base that modelizes one strategy for the access control of documents in the semantic wiki SweetWiki.

2.1 AMO Classes and Properties

AMO is based on some basic principles shared by all content management systems:

- *Agents* of a content management system are the users, user groups, services that interact with the system.
- These agents have *roles*. In the case of collaborative editing systems such as wikis or CMS, these roles are those of guest (agent not registered in the system), contributor, administrator. Other roles can be modeled depending on the kind of system.
- Each role is associated with a list of authorized *actions*. In the case of collaborative editing systems, the possible actions on a resource are creation, reading, modification and destruction of content, modification of access rights, modification of the list of agents allowed on a resource, change of the access type defined for a resource. Other actions can be modeled for other kinds of systems.
- There are different *types* of access to resources. We choosed to implement a strategy popular in some collaborative editing systems: a resource can be public (all users have reading and writing access), private (only authorized agents have reading and writing access) or semi-private (free reading access, writing access only to authorized agents). Again, other types of access can be added for other types of systems.
- Finally, the actions authorized to an agent on a resource depend on the role of the agent and/or the type of access defined for the resource.

The AMO ontology presented in Figure 1 provides the concepts necessary to represent this knowledge. The three classes `Role`, `Action` and `AccessType` are central to AMO. `Role` is the meta-class of classes `Administrator`, `Contributor` and `Guest`. `Action` is the meta-class of classes `ReadContent`, `ModifyContent`, `DeleteContents`, `ModifyUserRights`, `ModifyAccessType` and `ModifyAuthorizedAgents`. Finally, `AccessType` is the meta-class of classes `Private`, `Public` and `SemiPublic`.

Three classes of the FOAF vocabulary — the standard for social web discussed in section 4 — are also central in AMO: `Agent` and its sub-class `Group` and `Document`. They are used as domain or range of properties of AMO and also in the rules of AMO.

Properties `creator` and `hasAuthorizedAgent` associate an agent to a document (they have for domain the class `Document` and for range the class `Agent`); `hasRole` associates a role to an agent and `hasActionOnResource` an action to a role; property `hasAccessType` associates an access type to a document.

In addition, to represent into a model of binary properties the ternary relation which states that an agent is authorized to perform an action on a resource, we have

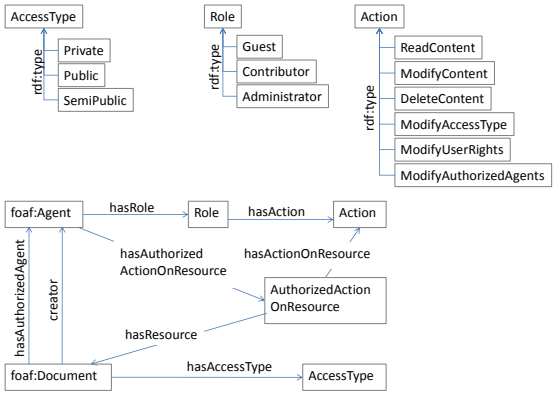

Fig. 1 AMO classes and properties

reified this relationship by introducing the subclass `AuthorizedActionOnResource` specializing the class `Action`, a property `hasAuthorizedActionOnResource` that associates an instance of `AuthorizedActionOnResource` to an agent, and the properties `hasDocument` and `hasAction` that associate to an instance of `AuthorizedActionOnResource` respectively a document and an action.

This RDFS vocabulary is both used to annotate the resources which we want to control access and to write inference rules modelling a chosen access policy.

2.2 AMO Inference Rules

Content management systems adopt access control strategies to resources that can vary from one system to another. Rather than varying the annotations of resources depending on the control strategies, we propose to model the control strategy declaratively: in the AMO ontology itself, as a base of inference rules. Some rules may vary depending on the strategy modeled while the annotations remain unchanged. Moreover, when compared to OWL, our choice of a base of rules combined with a light RDFS ontology enables a clear distinction between primitive concepts and rules and therefore an easy update or modification of the latters according to one strategy or another — without modifying the concepts. More generally, the advent of the new W3C standard RIF (standing for rule interchange format) offers a true alternative to OWL.

The rule base presented here is that of SweetWiki whose strategy of access control is similar to that of the widely used open source wiki Mindtouch Deki [2]. By default, administrators have all rights on all resources. The contributors have all rights relative to the content of resources, those reported as agents of a resource by

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

Fig. 2 An access control policy modeled in AMO

	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		

the author thereof also have some administrative rights on it. Guests are only allowed to read the content of resources. Figure 2 below summarizes the access rights to a resource depending on the type of access and the role of the user who tries to access the resource (horizontally are the types of access resources, vertically user roles).

We model this strategy in AMO *declaratively* by six inference rules, each corresponding to a situation described in Figure 2. For example, Rule 1 below specifies the rights granted to agents of a given resource. Other rules describe general laws such as *a member of a group inherits the roles assigned to her group* (Rule 2) or *the creator of a resource is an agent of this resource* (Rule 3).

These rules are expressed in the SPARQL language[3], by using the query pattern CONSTRUCT/WHERE: such a query enables to *construct* RDF graphs by replacing the variables of its clause CONSTRUCT by the values that satisfy the clause WHERE (they are retrieved by searching for potential matches to clause WHERE with the RDF data available in the content management system). A query CONSTRUCT/WHERE can therefore be seen as a rule applied in forward chaining, with clause WHERE the premise and clause CONSTRUCT the conclusion. These rules however can also be used in backward chaining, as it is the case in the Corese engine.

Rule 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 }
```

[3] Our rule base should be rewritten in the RIF format as soon as the semantic engine we use will handle this new standard.

Rule 2:

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

Rule 3:

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

This *declarative* modeling of the strategy of access rights management ensures easy maintenance. Changing rights of a class of users — and this for all resources involved — will only require the addition or deletion of triples statements in the conclusion of a rule. Similarly, the addition of new roles will only require the addition of a class representing this role and the rules representing the access rights associated with that role.

3 Access Rights Management in SweetWiki

The AMO ontology is used in the ISICIL project to annotate resources shared by a social network of business watchers. The management of access to these resources in the engine SweetWiki is based on (1) the exploitation of these semantic annotations, (2) inferences on these annotations based on AMO rules and (3) the formulation of SPARQL queries to retrieve knowledge about the authorized access to a specific user on a given resource. In SweetWiki, annotations of resources are based on FOAF, SIOC and AMO ontologies and SPARQL queries are used in most of the features implemented: RDF annotations feed the semantic engine Corese embedded in SweetWiki. In particular, by using the approximate search possibilities of Corese[3] and a system of semantic tagging of documents, SweetWiki offers an ”intelligent” browsing mechanism enhanced by suggestions.

3.1 Annotation of ISICIL Ressources with AMO

When creating a wiki page, the identity of its creator is registered and also the type of access to the page that is decided by her and possibly one or more agents authorized on the page, also designated by the creator. In SweetWiki this knowledge is represented into RDF annotations associated with the created pages. For example, Annotation 1 below results from the creation of a private wiki page by the user `AnnaKolomoiska` who stated that agent `MichelBuffa` is authorized on this page. This annotation uses the AMO properties `creator`, `hasAuthorizedAgent` and `hasAccessType` (and the class `WikiArticle` of the SIOC vocabulary discussed in 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>
```

When registering a user in SweetWiki, this information is represented in an RDF annotation. For example, Annotation 2 below states that `MichelBuffa` is a contributor to the wiki. It uses the AMO class `Contributor` and AMO property `hasRole` (and the class `Agent` of the FOAF vocabulary discussed in section 4).

Other annotations express knowledge relative to the user groups of the wiki. For example, Annotation 3 states that `AnnaKolomoiska` and `CatherineFaron` are members of the administrator group of the wiki. It uses for that the AMO property `hasRole` (and the FOAF classes `Group` and `Agent` and the FOAF property `member`).

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>
```

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 Inferences with the rule base of AMO

Applied to the annotations of the ISICIL resources, AMO rules enable to infer the rights of the wiki users on these resources. For example, consider again Rule 1 that illustrates section 2.2. Its premise pairs with Annotation 1 that illustrates Section 3.1: the resource `TestPage` is of type `WikiArticle` — a

class of the SIOC vocabulary, subclass of the class `Document` of the FOAF vocabulary — and `TestPage` is related to the user `MichelBuffa` with the `hasAuthorizedAgent` property. Applied on Annotation 1, Rule 1 allows to conclude that `MichelBuffa` has the *read*, *modify* and *delete* permissions on the content of the annotated resource `TestPage` and the *modify* permission on its type of access and its list of agents.

Similarly, Rule 2 applied on Annotation 3 allows to conclude that user `CatherineFaron` has the administrator role. Another rule of AMO (not provided here) describes general rights of an agent having the administrator role on any resource. It enables to conclude that `CatherineFaron` owns all the rights on the specific resource `TestPage`.

Finally, Rules 1 and Rule 3 applied on Annotation 1 enable to conclude that user `AnnaKolomoiska`, creator of resource `TestPage`, has the rights of an agent of that resource: *read*, *modify* and *delete* rights on its content and *modify* right on its type of access and its list of agents.

3.3 SPARQL Requests for Access Rights Management

Access to a particular resource by a given user depends, as all the actions in SweetWiki, on the answers to a SPARQL query provided by the Corese engine launched on the base of resource annotations. For this, Corese combines backward chaining on the AMO rule base and matching of queries with the annotation base. For example, the answer to the following SPARQL query will indicate whether the user `CatherineFaron` is allowed to modify the content of the resource `TestPage`:

Query 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> }
```

Other SPARQL queries are formulated to support all the functionalities of SweetWiki. For instance, the processing of the following query will provide the list of all the users having some rights on resource `TestPage` and for each of them it will state the list of her authorized actions on`TestPage`:

Query 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 Positioning

4.1 XML Languages for Access Control and Digital Rights

Most of the mechanisms of access control implemented in content management systems are based on XML languages dedicated to the description of policies of access control and digital rights management (DRM). These systems exploit the metadata associated with resources to which access must be controlled and these metadata comply with the XML schemas of these dedicated languages. Among these languages, the most famous are XrML[4] (Right eXtensible Markup Language) used as the basic language of expression rights of MPEG-21[5], ODRL[6](Open Digital Right Language) implemented by the Open Mobile Alliance (OMA) and XACML[7] (Extensible Access Control Markup Language) developed by OASIS. The ODRL model is based on the concepts of *Asset*, *Party*, *Permission*, *Constraint*, *Requirement*, *Condition*, *Rights holder*, *Context*, *Offer*, *Agreement* and *Revoking rights*. The XACML model allows to represent access control policies by rules. It is based on the concepts of *Rule*, *Policy* and *Policy Set* and these concepts can be refined with those of *Subject*, *Resource*, *Action*, *Environment*. A *Rule* comprises *Conditions* and *Effects* and a *Policy* embeds *Rules* and *Obligations*.

4.2 Semantic Approaches to Access Control

With the emergence of the Web of data and people, new approaches to manage access to content have emerged based on semantic Web models and technologies. Notably [4] shows the limitations of solutions using non-semantic description languages for managing access rights. They propose an OWL ontology to describe the access to web services inspired from the XACML model. More generally, in the few existing semantic models for managing access to content, we recognize some concepts that were already present in the older XML languages.

The W3C initiative is also noticeable: it uses since 2001 an RDF-based system to control access to the files of its servers: W3C ACL System[8]. [5] recently proposed an evolution of this system to a scalable system that allows for decentralized user authorization via an RDF metadata file containing an access control list (ACL). The ontology used in this system is called Basic Access Control Ontology[9]. It is presented as a basis to develop more sophisticated models. Unlike the AMO ontology presented in this paper, the authors do not propose a rule-based access control (al-

[4] http://www.xrml.org/

[5] http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm

[6] http://www.w3.org/TR/odrl/

[7] http://www.oasis-open.org/committees/xacml/

[8] http://www.w3.org/2001/04/20-ACLs

[9] http://www.w3.org/ns/auth/acl

though they suggest it as a possible future development), or a control at document level (rather than directory level).

Requirements in terms of access rights management in the ISICIL project are similar to those of digital libraries which [6] propose an overview. However, one of the key issues for digital libraries is not relevant in the context of ISICIL: the respect of the copyrights of available documents and for this purpose the protection of documents by DRM. Indeed the documents handled by the watchers remain in the corporate intranet or are public documents on the web. Among the work on access management in digital libraries, we notice those of [7] on the Fedora architecture for managing digital resources and those of [8] on the semantic Digital Library JeromeDL.

The Fedora authors propose a model called DARS (acronym for Distributed Active Relationships) for associating metadata to objects in a digital library, especially for managing access rights. However, although part of the model of access management is thus in an ontology, the Fedora system also uses XACML metadata associated with resources the it handles.

Access management in JeromeDL is based on the EAC ontology[10] (acronym for Extensible Access Control) [9]. EAC enables to associate licenses to resources, for each license corresponding to an access policy. For example, a license can specify that only people of a given organization can access some resources of the library. The purpose of EAC is to filter access to resources while that of AMO is to define access rights associated to user roles.

Approaches of access control based on annotations of resources are particularly well suited for social platforms. For example, in [10] end users are able to annotate by tagging both resources and members of their social network. Access control policies are then based on these annotations. For example, a basic policy states that if a resource shares the same tag as a member of the social network, this member has access to the resource. This user-centric approach is more flexible than role-centric access control since no real role nor actions need to be defined. It does not require an administrator user having global maitenance access rights to the system. Therefore, it seems more dedicated to the mananagement of personnal data rather than public shared data with many contributors (like in wikis). However, we plan to investigate how we could combine such a user-centric approach with our role-centric approach.

4.3 Social Web Standards

A key specificity of the approach we adopt with AMO is to be interoperable with the models of the social web and semantic web. Specifically, SweetWiki uses FOAF and SIOC concepts to annotate resources and AMO *complements* these ontologies to manage access to content (as show in section 2, some AMO properties have FOAF

[10] http://www.jeromedl.org/eac/1.0/spec/index.html

or SIOC classes for domain or range). FOAF[11] (acronym for Friend Of A Friend) is an RDF vocabulary used in social networks to describe people and and the relations among them. SIOC[12] (acronym for Semantically-Interlinked Online Communities) is another RDF vocabulary that models the concepts of social web applications: forums, blogs, wikis. It reuses some concepts from FOAF and other popular ontologies (Dublin Core, SKOS, etc.) and it has established itself as the standard. It is now integrated into numerous applications such as the WordPress blog engine and its adoption within the *Linked Data*[13] project confirms its popularity. A SIOC module was planned for some time for modeling access rights but it remained empty until recently. Now that it has been populated, we should state the alignment of AMO classes and properties with it. Briefly, the AMO classes `Role`, `Action` and `AccessType` should probably be aligned with those of `Role` in *SIOC Core Ontology* and `Permission` and `Status` newly added in the *SIOC Access Ontology* module.

FOAFRealm is an extension of FOAF proposed to collaboratively filter access to resources based on user profiles and their relationships in a social network. This vocabulary is used for example in JeromeDL for filtering based on measures of trust in a social network. Such filtering may be complementary to access control allowed by AMO, based on user roles and types of access to resources.

Finally, the problem of authorizing access to resources addressed in this paper is related to the problem of authentication of agents that will be tackled in the context of the ISICIL project. We intend for this to use the FOAF-SSL protocol[11] with which AMO is compliant — since it builds upon FOAF.

5 Conclusion and On-going Work

We have presented the AMO ontology dedicated to access management in content management systems. AMO comprises both (1) a set of classes and properties to annotate resources and thus control access to them by querying their annotations; and (2) a base of inference rules which declaratively represent a strategy for access control that can be easily modified — without affecting the annotations of resources —, depending on the strategy to implement on a particular web application. We have showed the use that can be made of AMO through examples of annotations of resources and rules in SweetWiki and we validated it by writing and processing SPARQL queries enabling to implement access control functionnalities by querying the base of annotations with the Corese semantic engine.

Prototyped in SweetWiki, the access management layer based on AMO will be implemented and deployed in the application framework of the ISICIL project, that involves different types of resources and applications, not only wikis. AMO is al-

[11] http://xmlns.com/foaf/spec/

[12] http://sioc-project.org/ontology

[13] http://linkeddata.org/

ready used in this framework, in the semantic network analysis and modeling module. Specifically, classes and properties of AMO occur in the annotations handled by the user profile server. The SemSNI (SEMantic Social Network Interactions) ontology used for these annotations provides a model of the interactions between users of a social network. It refers to AMO for defining access rights to shared resources for each user. SemSNI and AMO have also been used to specify access to documents shared by users of the Ipernity.com social network (a French site in the way of FaceBook specialized in photo sharing)[12].

More generally, AMO is based on FOAF and SIOC and is thus compatible with semantic web applications whose development is based on these popular vocabularies to describe their resources. AMO could therefore easily be integrated in such applications to support access control management.

Considering the state of the art, XML languages prior to the semantic Web, such as XACML requiere the support of large and complex software libraries to implement the management of rights while AMO remains simple to implement. Representing an access control policy by a rule base enables to overcome the implementation of complex mechanisms for calculating rights, which otherwise call for inheritance handling (e.g. inheritance of the roles associated to groups to which an agent belongs, union of actions authorized by these roles), for ordering operations depending on their occurence (e.g. deny and then authorize or the opposite), for handling implicit mechanisms (e.g. the creator of a resource automatically gets certain rights to this resource), etc..

When compared to other semantic approaches to access management, none of them uses rules; the authors of the ACL ontology have just raised such a possibility of extension. The representation of access control policies by rules is a key feature of AMO. It enables to easily adapt access rights management to any change of control policy — by modifying the only rules involved in that change and without affecting the annotations of the resources whose access is controled.

Finally, AMO is a declarative model which is simple to extend and this is one of our ongoing work. We intend to take into account in our model some characteristics specific to social networks, especially user profiles and relations and confidence and trust. We focused in this paper on role-centric access control but other types of annotation-based access control like [10] should be considered as well. We also envision to reuse the FOAF Realm ontology and to define rules based on confidence and trust measures and methods of propagation of confidence and trust in social networks.

Acknowledgement

This work is supported by the ISICIL projet funded by the French National Research Agency (ANR).

References

1. Buffa, M., Gandon, F., Erétéo, G., Sander, P., Faron, C.: Sweetwiki: A semantic wiki. Journal of Web Semantics **6**(1) (2008) 84–97
2. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C.: Querying the semantic web with corese search engine. In: 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, IOS Press (2004) 705–709
3. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F.: Searching the Semantic Web: Approximate Query Processing Based on Ontologies. IEEE Intelligent Systems **21**(1) (2006) 20–27
4. Alam, A., Subbiah, G., Thuraisingham, B., Khan, L.: Reasoning with Semantics-aware Access Control Policies for Geospatial Web Services. In: 3rd ACM Workshop On Secure Web Services, SWS 2006. (2006) 69–76
5. Hollenbach, J., Presbrey, J., Berners-Lee, T.: Using RDF Metadata to Enable Access Control on Social Semantic Web. In: International Semantic Web Conference, ISWC 2009. LNCS, Springer (2009)
6. Coyle, K.: Rights Management and Digital Library Requirements. Ariadne **40** (2004)
7. Lagoze, C., Payette, S., Shin, E., Wilper, C.: Fedora: an Architecture for Complex Objects and their Relationships. Int. J. on Digital Libraries **6**(2) (2006) 124–138
8. Kruk, S.R., Cygan, M., Gzella, A.: JeromeDL - Semantic and Social Technologies for Improving User Experience in Digital Libraries. In: World Wide Web Conference, WWW 2008, ACM (2008)
9. Kruk, S.R.: Extensible Access Control (EAC) Ontology Specification. (2008) DERI, http://www.jeromedl.org/eac/1.0/spec/index.html/.
10. Nasirifard, P., Peristeras, V., Hayes, C., Decker, S.: Extracting and Utilizing Social Networks from Log Files of Shared Workspaces. In: 10th IFIP Working Conference on Virtual Enterprises, PRO-VE 2009. (2009) 643–650
11. Story, H., Harbulot, B., Jacobi, I., Jones, M.: FOAF+SSL: RESTful Authentication for the Social Web. In: ESWC Workshop Trust and Privacy on the Social and Semantic Web, SPOT 2009. (2009)
12. Erétéo, G., Buffa, M., Gandon, F., Corby, O.: Analysis of a Real Online Social Network using Semantic Web Frameworks. In: 8th International Semantic Web Conference, ISWC 2009. LNCS, Springer (2009)