View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Unitn-eprints Research

Department of
Information Engineering
and Computer Science DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

ACCESS CONTROL VIA LIGHTWEIGHT ONTOLOGIES

Fausto Giunchiglia, Bruno Crispo and Rui Zhang

May 2011

Technical Report # DISI-11-467

Submitted to IEEE International Workshop on Semantics, Security and Privacy (ICSC 2011).

Access Control via Lightweight Ontologies

Fausto Giunchiglia, Bruno Crispo
DISI, University of Trento
Via Sommarive I-15
38050 Italy
Email: {fausto,crispo}@disi.unitn.it

Rui Zhang
CCST, Jilin University
Str. Qianjin 2699, Changchun
Pro. Jilin, China 130012
Corresponding Author Email:rui@disi.unitn.it

***Abstract*—The paper presents Relation Based Access Control $RelBAC$, a model and a logic for access control which models communities, possibly nested, and resources, possibly organized inside complex file systems, as lightweight ontologies, and permissions as relations between subjects and objects. $RelBAC$ allows us to represent expressive access control rules beyond the current state of the art, and to deal with the strong dynamics of subjects, objects and permissions which arise in Web 2.0 applications (e.g. social networks). Finally, as shown in the paper, using $RelBAC$, it becomes possible to reason about access control policies and, in particular to compute candidate permissions by matching subject ontologies (representing their interests) with resource ontologies (describing their characteristics).**

I. Introduction

Internet business patterns such as B2B, B2C, C2C are no longer high-tech terminologies but, rather, they represent everyday activities involving virtually everybody from producers to end customers. Businesses exchange information in addition to products via B2B networks; they sell products to customers via B2C networks and customers can even sell their own stuff to one another through C2C interaction patterns. Furthermore, customers are now able to provide feedbacks for quality and service; sales managers of large companies can distribute advertisements about new products or special offers to the vendors; service companies are able to publish new services through these online media; and so on. Thanks to the Web 2.0, eBusiness can enrich the traditional vending pattern with more active involvement of the involved actors.

However, Web 2.0 applications present new challenges for access control that can be exemplified as follows:

- The access control system must be capable of protecting various kinds of objects in largely different scales, possibly organized in complex directory structures.
- Permissions, access control rules and policies should be defined relatively independently so that the evolution of the social network has minimal impact on access control policies.
- Manual rule creation and management are time-consuming and error-prone to the exponentially increasing complexity of the knowledge base.

$RelBAC$ (for Relation Based Access Control) is a new model and a logic which has been introduced in [1] with the overall goal of dealing with the problem on access control in Web 2.0 applications. The first key feature of $RelBAC$ is that its access control models can be designed using entity-relationship (ER) diagrams. As such, they can be seamlessly integrated into the whole system and vary according to the scale of the business. The second feature, which motivates the name $RelBAC$, is that permissions can be modeled as relations, and differently from the state of the art, e.g., $RBAC$ [2], they can be manipulated as independent objects, thus achieving the requirements of modularity and flexibility described above.

In this paper we take a step further and show how, using $RelBAC$, social networks and object organizations can be modeled as lightweight ontologies (as defined in [3]), by exploiting the translation from classifications and Web directories to lightweight ontologies described in [4]. This in turn allows us to model permissions as Description Logic (DL) roles [5], access control rules as DL formulas, and policies as sets of DL formulas and, therefore, to reason about access control simply by using off-the shelf DL reasoners, thus addressing the last requirement described above.

The paper is organized as follows. Section II gives the model and the logic of *RelBAC*, Section III describes the usage of lightweight ontology in *RelBAC*, Section IV shows the reasoning with lightweight ontologies, Section V lists the related work and we conclude in Section VI.

II. $RelBAC$: Relation Based Access Control

A. The model

The $RelBAC$ model can be represented as an ER Diagram with the following components:

- `SUBJECT` (or `USER`): it is a set of subjects that intend to access some resources. 'IS-A' relations exist between sets of subjects. The largest subject set is the collection of all the possible subjects.
- `OBJECT`: it is a set of objects or resources that subjects intend to access. 'IS-A' relations exist between sets of objects. The largest object set is the collection of all the possible objects of the system.
- `PERMISSION`: it is allows an operation that subjects can perform on objects, denoted by the name of the operation it refers to, e.g., *Write* or *Read*. A `PERMISSION` is a *relation* between `SUBJECT` and `OBJECT`, namely a set of (subject, object) pairs. 'IS-A' relation also exist between permissions.
- `RULE` (short for `ACCESS CONTROL RULE`): it associates a `PERMISSION` to a specific set of

(SUBJECT,OBJECT) pairs which assigns the specific SUBJECT the access right named by the PERMISSION onto the specific OBJECT. Rules are formalized as DL formulas, as described in the following subsection.

B. *The Logic*

The ER model of *RelBAC* can be directly expressed in DL. In general, SUBJECTs, and OBJECTs are formalized as concepts and PERMISSIONs are formalized as DL roles[1]. Individual SUBJECTs and OBJECTs are formalized as instances and PERMISSIONs are pairs of instances i.e. (SUBJECT, OBJECT). RULEs express the kind of access rights that SUBJECTs have on OBJECTs and are formalized as the *subsumption* axioms provided below. In Rules 6 and 12, we abbreviate $\forall\neg P.\neg O$ as $\forall O.P$, which allows us to assign a permission P to *all* objects in O. Thus, we may have a single subject 'u' having access to a single object (Rule 7), to some objects (Rule 8), to only the objects in O (Rule 9), to minimum or maximum n objects (Rules 10 and 11), or to all objects in a set O (Rule 12). Dual arguments can be given for any set of users 'U' by looking at the rules on the left (Rule 1 - 6). We call these rules *user-centric*, as they allow us to assign users fine-grained permissions such as those listed above. Dually, we can define corresponding *object-centric* rules by replacing *U, O, P, u, o* respectively with *O, U,* P^{-1}, *o, u*. This feature, not discussed here for lack of space, is however quite important in terms of access control as it allows to design policies from different perspectives.

$$\begin{array}{llll}
U \sqsubseteq P : o & (1) & (P : o)(u) & (7) \\
U \sqsubseteq \exists P.O & (2) & (\exists P.O)(u) & (8) \\
U \sqsubseteq \forall P.O & (3) & (\forall P.O)(u) & (9) \\
U \sqsubseteq \geq nP.O & (4) & (\geq nP.O)(u) & (10) \\
U \sqsubseteq \leq nP.O & (5) & (\leq nP.O)(u) & (11) \\
U \sqsubseteq \forall O.P & (6) & (\forall O.P)(u) & (12)
\end{array}$$

From the above, *RelBAC* shows a rich set of policy styles, which is even more articulated with the *object-centric* rules as described in [6]. In practice, the most commonly used assignments are the first and the last, which resemble the only two kinds of assignments allowed in *RBAC*. Subsumption is not only used to express access control RULEs but also used to represent the partial order '$\geq$' among subjects, among objects and among permissions. The ordering relation '$\geq$' translates the 'IS-A' relation in the *RelBAC* model and it allows us to build inheritance hierarchies among subjects, objects and permissions. Inheritance is a very valuable property as it largely simplifies the otherwise very complex task of administration [2]. We define '$\geq$' as follows:

$$U_i \geq U_j \quad iff \quad U_i \sqsubseteq U_j \tag{13}$$
$$O_i \geq O_j \quad iff \quad O_i \sqsubseteq O_j \tag{14}$$
$$P_i \geq P_j \quad iff \quad P_i \sqsubseteq P_j \tag{15}$$

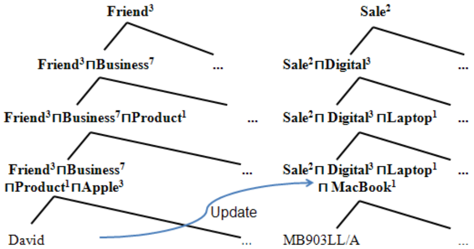
Friend^3
Friend^3⊓Business^7
Friend^3⊓Business^7⊓Product^1
Friend^3⊓Business^7 ⊓Product^1⊓Apple^3
David
...
Update
Sale^2
Sale^2⊓Digital^3
Sale^2⊓ Digital^3 ⊓Laptop^1
Sale^2⊓ Digital^3 ⊓Laptop^1 ⊓ MacBook^1
MB903LL/A

Fig. 1. Permission Assignment on Lightweight Ontologies

III. Lightweight Ontologies for Access Control

With the communication simplified by the development of Internet, social activities such as online forums and blogs greatly increase the number and type of relations in a social network: not only traditional relations like 'knows', 'is-a-friend-of', etc. but new terms such as 'shares-photo-with' or 'comments-on-blog'. In another perspective, people are familiar with tree-like structures such as the file systems of their computers, their email directories, classifications, catalogs, and so on. In general, there is a widespread tendency towards organizing resources in tree-like structures. The key feature underlying the success of tree-like directories is that one can easily find something according to the property that, the deeper a category is in a tree, the more specific resources it will contain. Thus, community access control can be implemented in *RelBAC* with the subjects, objects and permissions encoded into different lightweight ontologies. Our solution is, therefore to translate, with no or very little user intervention, these tree-like knowledge structures into lightweight ontologies. We achieve this goal by exploiting the ideas described in [7], in which the authors show how a classification or a Web directory can be automatically translated into a lightweight ontology. Any classification or directory where each category is labeled with a natural language name expressing its contents, can be translated into a lightweight ontology according to two main steps, as follows:

1) The label of each node is transformed into a propositional DL formula using natural language processing (NLP) techniques. For example, a label 'Soccer Fan' is transformed into '$Soccer^i \sqcap Fan^j$' where the superscript $i(j)$ stands for the ith (jth) meaning of the word in a reference dictionary (e.g., WordNet).
2) Each node is associated a formula, called the *concept at node*, which is the conjunction of the formulas of all the nodes on the path from the root to the node itself. For example, a node labeled 'Soccer Fan' will be labeled with '$Friend^k \sqcap Soccer^i \sqcap Fan^j$'. The concept at node univocally defines the 'meaning of that node', namely, the set of documents which can be classified under it.

The result of the two steps above is a lightweight ontology where each node is labeled with its concept at node and where each concept at node is subsumed by the concepts of all the nodes above. This property allows for automated object classification and query answering. People will keep seeing

[1] A DL role is a binary relation, not to be confused with a 'role' of the *RBAC* model.

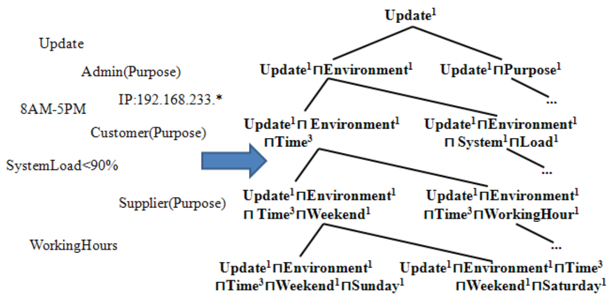
Update
Admin(Purpose)
IP:192.168.233.*
8AM-5PM
Customer(Purpose)
SystemLoad<90%
Supplier(Purpose)
WorkingHours
$Update^1$
$Update^1 \sqcap Environment^1$
$Update^1 \sqcap Purpose^1$
...
$Update^1 \sqcap Environment^1 \sqcap Time^3$
$Update^1 \sqcap Environment^1 \sqcap System^1 \sqcap Load^1$
...
$Update^1 \sqcap Environment^1 \sqcap Time^3 \sqcap Weekend^1$
$Update^1 \sqcap Environment^1 \sqcap Time^3 \sqcap WorkingHour^1$
...
$Update^1 \sqcap Environment^1 \sqcap Time^3 \sqcap Weekend^1 \sqcap Sunday^1$
$Update^1 \sqcap Environment^1 \sqcap Time^3 \sqcap Weekend^1 \sqcap Saturday^1$

Fig. 2. Scattered Permissions to a Lightweight Ontology

and managing a classification but all their operations will be supported and (partially) automated via the background reasoning operating on the underlying lightweight ontology. This background ontology has the same (tree-like) structure as the original classification, but it makes explicit, with its 'IS-A' hierarchy, all the originally implicit and ambiguous relations between object categories. This substantially contributes to address the access control problem. More concretely, some advantages are:

- Objects can be automatically classified into the proper directories with the help of a DL reasoner. By exploiting the ideas described in [7] it becomes possible to easily add the vast amount of new information to the proper categories with the proper access rules;
- The evolution of the object ontology (e.g., addition or deletion of a category) is much more under control because it must satisfy the underlying ontological semantics;
- With the partial order formalized as in Section II-B, the permissions on an object category will propagate up the tree without extra policies (discussed more in Section IV).

Considerations similar to those provided for object ontologies apply also to subject ontologies. As mentioned above, these ontologies can be used to organize access to the underlying (possibly very messy) social network. There are however two further important considerations. The first is that $RelBAC$ subject lightweight ontologies closely resemble $RBAC$ role hierarchies [2]. They are however easier to manage as users and permissions are totally decoupled. The second is that the links across subjects in a social network, can be used to suggest candidate paths for permission propagation. One such small example is depicted in Figure 1.

Finally, the translation into a lightweight ontology can be applied also to permission hierarchies. Notice that natural language labels have been translated into DL formulas. The terms on the left of Figure 2 are meant to provide evidence of how the step from natural language to logic allows us to organize otherwise sparse categories. Notice how the lightweight ontology in Figure 2 is upside down with respect the object and subject ontologies presented before. In particular the top category is the most powerful and less populated (in the sense that it is the one satisfied by the smallest number of subject object pairs). This notation is quite common in access control and it satisfies the intuition that the categories corresponding to the highest number of permissions should be put at the top of the hierarchy.

IV. Reasoning about Access Control Rules

The management and administration of access control with complex subject, object and permission structures are quite challenging and error-prone. In $RelBAC$, by exploiting the translation into lightweight ontologies described in Section III, these activities can be strongly supported by providing tools (i.e., DL reasoners) which automate much (if not all) of the reasoning about access control such as design time ontology consistency checking, permission propagation management, separation of duties, etc. Some examples of reasoning are:

Design Time Consistency Checking It is almost impossible to check manually a large access control knowledge base, not to say further integration of multiple knowledge bases. The reasoning service of $RelBAC$ offers consistency checking such as to check if $\mathcal{S} \cup \mathcal{P} \models \bot$, where $\mathcal{S}, \mathcal{P}$ stand for the knowledge bases corresponding to the state description and policy description. If the answer is negative, the knowledge base is consistent.

Permission Propagation An advantage of the hierarchy formalized as 'IS-A' relations through subjects, objects and permissions provide 'free' permission propagation by the reasoning. For example, in the predefined knowledge base we know 'Bob is a business friend', 'write is more powerful than read', 'laptop is a subset of digital device'. Thus we can reason the permission propagation as $\{Bussiness(Bob), Write \sqsubseteq Read, Laptop \sqsubseteq Digital, Business \sqsubseteq \forall Digital.Write\} \models (\forall Digital.Write)(Bob)$.

Separation of Duties (SoD) To enforce that some permissions should not be assigned to some users at the same time is the basic idea of *SoD*. For example, 'customers should not be allowed to read and update some category, say Player'. And it's straight forward to be secured by a rule in the knowledge base as $(Update : Player) \sqcap (Read : Player) \sqcap Customer \sqsubseteq \bot$.

Access Control Decision At run time, the access control system will face various of access control requests and make decisions at real-time. $RelBAC$ turns a request into a formula and put it to the reasoner and then the reasoner will check whether it is consistent with the knowledge base. A positive answer means that the request is acceptable, otherwise should be denied.

However the fact that we handle subject, object and permission hierarchies as lightweight ontologies allows us to deal with the problem of semantic heterogeneity, namely with the fact that in general we will have multiple subject and/or object and/or permission hierarchies which express semantically related notions in many different forms. This problem has been addressed as *semantic matching* in [8]. In the domain of access control this problem becomes quite relevant as we see two kinds of applications of the semantic matching techniques.

1) Two hierarchies of the same kind such as two subject hierarchies, two object permission hierarchies, etc.

2) One subject and one object hierarchy. We found that there exists similarity between the subject and object lightweight ontology although they are heterogeneous ontologies built independently.

Let's go back to Figure 1, it shows parts of the lightweight ontologies built on two hierarchies, one subject and one object. On the left, David is classified as an instance of the set '$Friend^3 \sqcap Business^7 \sqcap Product^1 \sqcap Apple^3$' according to his social position that he has a $Business^7$ relation with Alice and he works for $Apple^3$ (which is an IT company rather than a fruit). On the right, there's a class of objects '$Sale^2 \sqcap Digital^3 \sqcap Laptop^1 \sqcap MacBook^1$' where $Sale^2$ is a branch of $Business^7$, $MacBook^1$ is a $Laptop^1$ as a $Product^1$ of $Apple^3$. Apparently the two concepts are different in labels, but semantically overlapping.

To detect semantic relations between lightweight ontologies, we use S-Match as described in [8]. The original idea is to calculate the semantic similarity such as *equal, overlapping*, etc. between the categories of the two given classifications, such as the subject, object or permission hierarchies, when two organization integrates and verify that some desired properties (i.e. SOD) still hold.

V. Related Work

Classic access control techniques, e.g., cryptography have been proposed for community access control such as [9], [10]. Such systems focus on protection from security threats rather than taking use of the rich information from the web. Lockr[11] was proposed to fit the situation that the large number of content sharing systems and sites use different access control methods un-reusable for each other. It separates social networking information from the content sharing mechanisms, so that end users do not have to maintain several site-specific copies of their social networks. It also provides a way to use social relationships as an important attribute, *relationship type*, to define access control rules.

Another series of research focus on providing policy languages for the rich semantics on the web. Yague et al. in [12] even presented a model named Semantic Based Access Control. The model is based on the semantic properties of the resources, clients (users), contexts and attribute certificates and relies on the rich expressiveness of the attributes to create and validate access control policies. Dimiani et al. proposed in [13] to exploit context information in Web-based environment access control, the context information formalized in DL can be used as preconditions of *RelBAC* rules. Pan et al. present a middle-ware based system [14] to use semantics in access control based on the $RBAC$ model [2] with a mediator to translate the access request between organizations by replacing roles and objects with matched roles and matched objects. They used semantic mapping on roles in order to find the similarity or separation of duties between roles in two ontologies. We do further as the S-Match tools are more generic and can match a subject ontology with an object ontology in order to suggest new rules.

VI. Conclusion

In this paper we have presented $RelBAC$, a new model and logic for access control. The main feature of $RelBAC$ is that it allows to organize users and objects as (lightweight) ontologies and that it models permissions are relations. This in turn allows to represent access control rules and policies as DL formulas and therefore to reason about them using state of the art off-the-shelf reasoners. In turn, as shown in the second part of the paper, this allows us to match, using the semantic matching technology, the user and the object ontologies and, as a consequence, to generate (semi-)automatically permissions which (may) fit the user interests. The idea is that these permissions are then proposed to the administrator as suggestions to be confirmed and approved.

References

[1] F. Giunchiglia, R. Zhang, and B. Crispo, "Relbac: Relation based access control," in *International Conference on Semantics, Knowledge and Grid, SKG 2008*, I. C. Society, Ed., 2008.

[2] D. F. Ferraiolo, R. S. Sandhu, S. I. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[3] F. Giunchiglia and I. Zaihrayeu, *Encyclopedia of Database Systems*. Verlag, Springer, June 2009, no. 978-0-387-35544-3, ch. Lightweight Ontologies.

[4] F. Giunchiglia, M. Marchese, and I. Zaihrayeu, "Encoding classifications into lightweight ontologies." in *ESWC*, 2006, pp. 80–94.

[5] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The description logic handbook: theory, implementation, and applications*. New York, NY, USA: Cambridge University Press, 2003.

[6] R. Zhang, *Relation Based Access Control*. Netherlands: IOS Press, 2010.

[7] F. Giunchiglia, M. Marchese, and I. Zaihrayeu, "Towards a theory of formal classification," in *CandO 2005,AAAI-05*, Pittsburgh, Pennsylvania, USA, 2005.

[8] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, "Semantic matching: Algorithms and implementation." *J. Data Semantics*, vol. 9, pp. 1–38, 2007. [Online]. Available: http://dblp.uni-trier.de/db/journals/jods/jods9.html\#GiunchigliaYS07

[9] B. Carminati and E. Ferrari, "Privacy-aware collaborative access control in web-based social networks." in *DBSec*, ser. Lecture Notes in Computer Science, V. Atluri, Ed., vol. 5094. Springer, 2008, pp. 81–96. [Online]. Available: http://dblp.uni-trier.de/db/conf/dbsec/dbsec2008.html\#CarminatiF08

[10] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "A semantic web based framework for social network access control," *Proceedings of the 14th ACM symposium on Access control models and technologies SACMAT 09*, pp. 177–186, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1542207.1542237

[11] A. Tootoonchian, K. K. Gollu, S. Saroiu, Y. Ganjali, and A. Wolman, "Lockr: social access control for web 2.0," in *WOSP '08: Proceedings of the first workshop on Online social networks*. New York, NY, USA: ACM, 2008, pp. 43–48.

[12] M. I. Y. del Valle, M. del Mar Gallardo, and A. Mana, "Semantic access control model: A formal specification," in *ESORICS*, ser. LNCS, S. D. C. di Vimercati, P. F. Syverson, and D. Gollmann, Eds., vol. 3679. Springer, 2005, pp. 24–43. [Online]. Available: http://dblp.uni-trier.de/db/conf/esorics/esorics2005.html\#ValleGM05

[13] E. Damiani, D. Capitani, C. Fugazza, and P. Samarati, "Extending context descriptions in Semantics-Aware access control," 2006, pp. 162–176. [Online]. Available: http://dx.doi.org/10.1007/11961635_11

[14] C.-C. Pan, P. Mitra, and P. Liu, "Semantic access control for information interoperation," in *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*. New York, NY, USA: ACM, 2006, pp. 237–246.