Wireless Pers Commun (2010) 55:5–17
DOI 10.1007/s11277-009-9782-4

Relation-Based Access Control: An Access Control Model for Context-Aware Computing Environment

Rui Zhang · Fausto Giunchiglia · Bruno Crispo · Lingyang Song

Published online: 5 August 2009

Abstract Context-aware computing is an important aspect of the pervasive computing environment and its various dynamic context information brings new challenges to access control systems. In this paper a new access control model, relation based access control (*RelBAC*), is provided for context-aware environment with a domain specific Description Logic to formalize the model. The novelty of *RelBAC* is that permissions are formalized as binary relations between subjects and objects which could evolve with the dynamic contexts. The expressive power of *RelBAC* is illustrated in a case study of a project meeting event.

Keywords Access control · *RelBAC*

1 Introduction

A context-aware system is usually regarded as a component of a ubiquitous computing or pervasive computing environment. Context-aware computing has been brought to us by the evolution of wireless technologies such as bluetooth, global positioning system (GPS) and

This work was done during the PhD program of the first author in University of Trento.

R. Zhang (✉)
College of Computer Science and Technology,
Jilin University, Street Qianjin, Changchun 2699, China
e-mail: zhangrui326@gmail.com

F. Giunchiglia · B. Crispo
DISI, University of Trento, Via Sommarive 14, Povo, Trento 38050, Italy
e-mail: fausto@disi.unitn.it

B. Crispo
e-mail: crispo@disi.unitn.it

L. Song
Peking University, Beijing, China
e-mail: lingyang.song@pku.edu.cn

smart agents. It offers much easier ways to access resources. Thus comes the issue of access control: how to protect the resources in a context-aware system?

Although several context definitions have been proposed in various papers such as [2,16], a broad notion of context in a pervasive computing environment is the characters of ambient conditions and physical world situations which is relevant to performing appropriate actions in the domain. The context of a person may consist of the physical location, the portable devices, the network s/he is involved, etc. Environment attributes such as system load, connection number, etc., build the context of the resources. Temporal attribute is a special context which could be the character of not only the user, the resource but the access activity as well. Thanks to the development of sensors and geographic information system (GIS) technologies, various, changeable and vast amount of context information can be collected in a context-aware computing system. This requires an access control mechanism with the following characteristics:

- There should be a formal representation to describe the context aware system such that formal method of validation and reasoning can be performed. As the amount of context information increases every millisecond, simply manual administration is almost impossible. A computer aided administration scenario becomes more and more compulsory for access control in a pervasive environment.
- It should be adjustable to various contexts, personal preference and domain requirements. With the research going on in pervasive computing, context-aware systems will be applied in diverse fields. A fixed model cannot catch all the aspects of a pervasive environment, not to say more complex domains such as time which is not completely studied.
- It should be easy to embed into smart agents or low performance devices. As a pervasive computing system aims at information access at anywhere at any time, capabilities of the devices should be paid plenty of attention from an efficiency point of view.

In this paper, a refined access control model, relation based access control (*RelBAC*), is described together with the logical formalization. Considering the requirements above, *RelBAC* has the following properties.

- *RelBAC* models the permissions of a subject onto an object as *relations*, which makes permissions as a first class entity in the model. The abstraction of a permission as $(subject, object)$ pair is natural with respect to human intuition. Moreover, since permissions are regarded as a first class component, it is easy to analyze the characteristics of a permission and the relations between permissions, such as to analyze the effects of contexts on a permission and the interaction between permissions.
- *RelBAC* offers a set of logics based on Description Logic which has a compact language and mature reasoners. Given the very same set of subjects S and set of objects O, the rich expressiveness of *RelBAC* logic allows to assign different permissions such as to grant each subject in S access to *some, only, minimum/maximum n* or *all* objects in O. Similar assignments can be made from the object point of view.
- *RelBAC* offers an access control model with a simple extensible entity relationship diagram which can be easily integrated into the system design. Thanks to the extensibility, *RelBAC* can easily capture various contexts such as location attribute and related context with the compulsory components such as subject, object and permission.

The paper is organized as follows. Section 2 gives a short survey of the related work that inspires our model; Sect. 3 describes the *RelBAC* model; Sect. 4 shows the *RelBAC* logic and how to translate permission assignment into *RelBAC* axioms; Sect. 5 presents a conference event case study of *RelBAC*; and we conclude in Sect. 6.

Fig. 1 The ER Diagram of the *RelBAC* Model

2 Related Work

Context aroused the interest of access control researchers many years ago. Many researchers extended existing access control models such as RBAC [7] to support context information such as in [4,11,17]. Environment contexts such as time and system information were modeled as environment role by Moyer et al. [17] together with the extension to support object role. Joshi and Bertino et al. studied the time context and its impact on access control in [11] and offered a temporal RBAC model focusing on temporal constraints. Space related context was studied by Damiani and Bertino et al. [4]. Covington et al. proposed a context-dependent policy model in [3] where context is modeled with environment role to restrict and regulate user privileges, but predefined static $(attribute,\ constant)$ value pairs cannot adapt for evolving policy and there were no reasoning mechanism for the context and policies.

With the study in the importance of temporal, spatial and environment context especially in a pervasive computing environment where enormous amount of context information can be used on access control. Predefined roles are used in [10] to assure context information transmission between pervasive nodes. Zhang et al. [19] proposed dynamical adjust static role and permission assignments based on context information but they depend on a central authorizer to change the active role of user's agent according to the change of context. Another RBAC based context-aware model was proposed by Emami et al. [6]. It implements a two step access control with the user session registration to the domain authority and the session agent self govern of access with the session permission assignment database. Kulkarni et al. concentrate on constraints in [14] with a programming framework based on XML for building context-aware applications and a context guard mechanism to revoke role operation sessions. Both work in [6,14] are expressive in dynamic revocation, but are also application specific that are not easily adjust to evolving contexts.

Another direction focuses on providing a policy language for the environment such as in KAos [18], Rei [12,13] and Ponder [5]. From the reasoning ability point of view, KAos is the best as Description Logic has not only more expressive power but also many efficient reasoners.

3 The *RelBac* Model

A new access control model called *RelBAC* (Relation Based Access Control) appears in [9]. A revised model is shown in Fig. 1 where `SUBJECT`, `OBJECT` are entity sets and `PERMISSION` is relation set.

`SUBJECT`: A subject is an access requester, and a set of subject will be called a *group* later.

`OBJECT`: An object is a resource under access control, such as hard disk files, web tags, computers in GRID, etc. A set of object will be called a *class*.

`PERMISSION`: A permission is a named ordered pair $(s,\ o) \in \texttt{SUBJECT} \times \texttt{OBJECT}$ with a name like 'P' as the operation that s can perform on o. For example '$(rui, code1.0) \in$

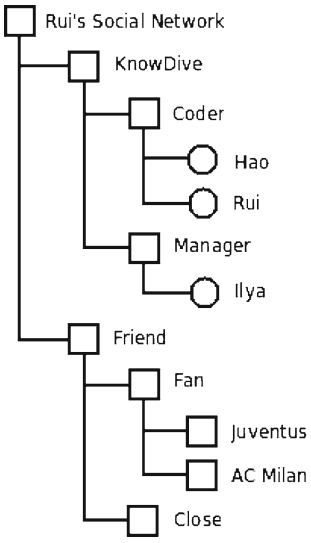


Fig. 2 SUBJECT Hierarchy: Rui's social network consists of mainly two groups, one from KnowDive research team and the other as friends. Individual subject such as 'Hao' is classified as a coder

Read' means intuitively that a SUBJECT 'Rui' represented by *rui* can access an OBJECT *code*1.0 with the operation *Read*.

IS-A: The loops on SUBJECT, OBJECT and PERMISSION represent 'is-a' relations. An IS-A relation between SUBJECT groups (OBJECT classes) is a many-to-many mapping among groups/classes, e.g., $(Coder, KnowDive) \in$ IS-A shows that according to some social network, *Coder* is a sub-group of *KnowDive*. An IS-A relation between PERMISSIONs is also a many-to-many mapping between sets of pairs where $(Write, Read) \in$ IS-A means that a pair (s, o) in set *Write* is also pairs in *Read*.

RULE (short for access control rule): A RULE associates a PERMISSION to a specific set of (SUBJECT, OBJECT) pairs by two means, *subject-centric* and *object-centric*. A *subject-centric* rule states that a set of SUBJECTs is a subset of those that can access given set of OBJECTs with a given PERMISSION; an *object-centric* rule states that a set of OBJECTs is a subset of those that can be accessed by a given set of SUBJECTs with a given PERMISSION. Thus *RelBAC* allows us to define rules from either the subject or the object stand. For example, in Linux system, access rights are attached to files which makes it easy to check the accessibility of a user, given a file URI. But it is hard to get all the users that have some access to a given file. *RelBAC* offers the *object-centric* rule which makes this as easy as to look from the object column in an access control matrix [15].

The scenario of personal desktop file access control is used as an example of *RelBAC*. Figures 2, 3 and 4 are examples of SUBJECT, OBJECT and PERMISSION hierarchies, respectively, where squares are sets and circles are individuals.

Now comes something more interesting that permissions may take the form of a tree (Fig. 4a) and an inverted tree (Fig. 4b). By Fig. 4a we state that to write or delete a file is a more specific (intuitively more powerful) operation than read and to update is even more specific. The circle denoting the pair $(Ilya, Code1.0)$ under 'Update' represents that Ilya can update the Code1.0 and the pair at bottom of Fig. 4a tells us that Rui can write Code1.0 and Hao can read (view) the video named 'Shrek II'. Figure 4b shows us that permissions may take the form of an inverted tree. Environment contexts such as accessing time and system load are important standards for access control and we can have various of

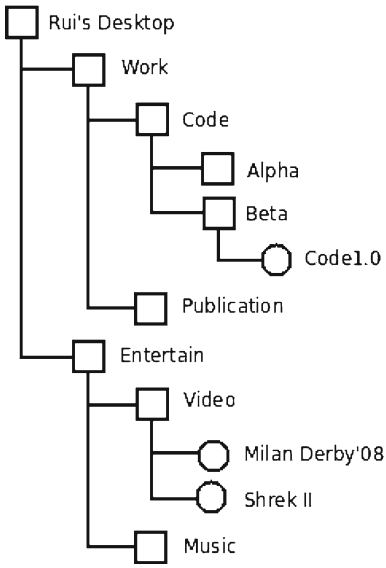


Fig. 3 OBJECT Hierarchy: Rui's working stuffs such as code and publications are organized in a tree structure as well as other files for entertainment. Individual files can be classified under the class such as '*Shrek II*' under 'Video'

permissions defined accordingly. These permission hierarchies might be different even for the same environment by different administrators, for example one may consider system load first and anther emphasizes accessing time instead. Actually, Fig. 4a is a Directed Acyclic Graph as the two square labeled 'Update' are the same. Therefore Ilya is allowed to update, delete and read Code1.0, and also perform write on it.

4 The *RelBac* Logic

RelBAC is described with an ER diagram in Fig. 1 not only because of the prevailing usage of ER diagrams in system design, but also because the inner relation between ER diagram and description logic (DL). In [1], Sattler et al. show a straight forward translation without losing any semantics. This brings a decidable logical framework with powerful reasoning ability which will automate the administration of access control.

Therefore, a family of Description Logics are used for the *RelBAC* model, named also as *RelBAC*. It is not a single logic as different access control rules corresponds to different kinds of constructors and leads to different complexity.

4.1 Language and Semantics

Here a $\mathcal{ALC}$ based Description Logic, $\mathcal{ALCQIBO}$, is used as the *RelBAC* logic. A more detailed discussion of $\mathcal{ALCQIBO}$ is in [20]. Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be pairwise disjoint and countably infinite sets of *concept names*, *role names* and *individual names*. Then *concept expressions* and *role expressions* are defined as follows:

$$C, D \ ::= \ A \ \mid \ \neg C \ \mid \ C \sqcap D \ \mid \ \geq n \, R.C \ \mid \ \{a_i\}$$
$$R, S \ ::= \ P \ \mid \ R^{-} \ \mid \ \neg R \ \mid \ R \sqcap S$$

where $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$, $a_i \in \mathsf{N_I}$ and $n \in \mathbb{N}$.

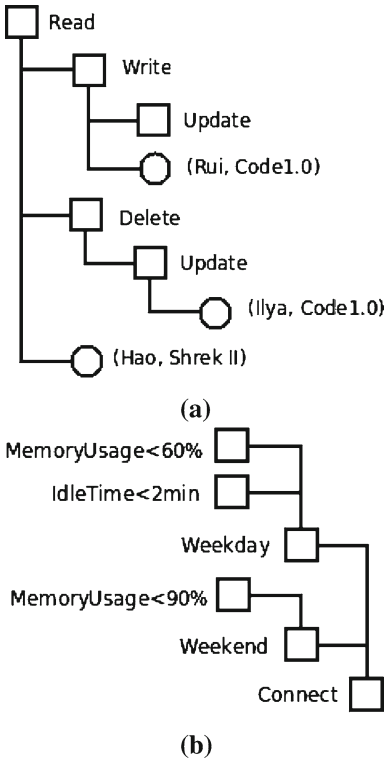


Fig. 4 PERMISSION Hierarchy: permissions can be organized as a tree, inverted tree or even a DAG

A knowledge base (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$, called *TBox*, is a finite set of *general concept inclusions (GCIs)* of the form $C \sqsubseteq D$ and a finite set of *general role inclusions (GRIs)* of the form $R \sqsubseteq S$, while $\mathcal{A}$, called *ABox*, is a finite set of concept and role assertions of the form $C(a_i)$ and $R(a_i, a_j)$, with $a_i, a_j \in \mathsf{N_I}$. An $\mathcal{ALCQIBO}$*-interpretation*, $\mathcal{I}$, is a pair $(\Delta, \cdot^{\mathcal{I}})$ where Δ is a non-empty set called the *domain* of $\mathcal{I}$ and $\cdot^{\mathcal{I}}$ is a function mapping each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta$ and each $P \in \mathsf{N_R}$ to a relation $P^{\mathcal{I}} \subseteq \Delta \times \Delta$. Furthermore, $\cdot^{\mathcal{I}}$ applies also to individuals by mapping each individual name $a_i \in \mathsf{N_I}$ into an element $a_i^{\mathcal{I}} \in \Delta$ such that $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for all $i \neq j$, i.e., we adopt the so called *unique name assumption* (UNA). We extend the mapping $\cdot^{\mathcal{I}}$ to complex roles and concepts as follows:

$$
\begin{aligned}
&(R^-)^{\mathcal{I}} := \left\{(y, x) \in \Delta \times \Delta \mid (x, y) \in R^{\mathcal{I}}\right\},\\
&(\neg R)^{\mathcal{I}} := \Delta \times \Delta \setminus R^{\mathcal{I}}, \quad (\neg C)^{\mathcal{I}} := \Delta \setminus C^{\mathcal{I}},\\
&(R \sqcap S)^{\mathcal{I}} := R^{\mathcal{I}} \cap S^{\mathcal{I}}, \quad (C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}},\\
&(\geqslant n\ R.C)^{\mathcal{I}} := \left\{x \in \Delta \mid \sharp\{y \in \Delta \mid (x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\right\}, \quad \{a_i\}^{\mathcal{I}} := \left\{a_i^{\mathcal{I}}\right\}.
\end{aligned}
$$

An $\mathcal{ALCQIBO}$-interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ is said a *model* of a KB, $\mathcal{K}$, iff it satisfies $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, for all $C \sqsubseteq D \in \mathcal{K}$, $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$, for all $R \sqsubseteq S \in \mathcal{K}$, $a_i^{\mathcal{I}} \in C^{\mathcal{I}}$, for all $C(a_i) \in \mathcal{A}$, and $\left(a_i^{\mathcal{I}}, a_j^{\mathcal{I}}\right) \in R^{\mathcal{I}}$, for all $R(a_i, a_j) \in \mathcal{A}$. In this case we say that $\mathcal{K}$ is satisfiable and write $\mathcal{I} \models \mathcal{K}$. A concept C (role R) is *satisfiable w.r.t.* $\mathcal{K}$ if there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that $C^{\mathcal{I}} \neq \emptyset$ ($R^{\mathcal{I}} \neq \emptyset$). A number of useful abbreviations are as follows:

$$
\begin{aligned}
C \sqcup D &\text{ for } \neg(\neg C \sqcap \neg D)\\
(\leqslant n\ R.C) &\text{ for } \neg(\geqslant n+1\ R.C)\\
(= n\ R.C) &\text{ for } \neg(\geqslant n+1\ R.C) \sqcap (\geqslant n\ R.C)\\
\exists R.C &\text{ for } (\geqslant 1\ R.C)\\
\forall R.C &\text{ for } (\leqslant 0\ R.\neg C)\\
\top &\text{ for } A \sqcup \neg A \text{ (for some concept } A)\\
\bot &\text{ for } \neg\top\\
\forall C.R &\text{ for } \forall\neg R.\neg C
\end{aligned}
$$

4.2 Hierarchy Representation

RelBAC uses *subsumption* axioms to represent the `IS-A` hierarchy among subjects, among objects and among permissions. For example in Figs. 2, 3 and 4 we have relations as follows.

$$Coder \sqsubseteq KnowDive \qquad Code \sqsubseteq Work$$
$$Write \sqsubseteq Read \qquad Connect \sqsubseteq Weekends$$

In general, hierarchy of subjects, objects and permissions can be defined by partial order '$\geq$' representing the `IS-A` relations.

$$S_1 \geq S_2 \textit{ iff } S_1 \sqsubseteq S_2, \quad O_1 \geq O_2 \textit{ iff } O_1 \sqsubseteq O_2, \quad P_1 \geq P_2 \textit{ iff } P_1 \sqsubseteq P_2.$$

$S_1 \geq S_2$ means S_1 is a subject group more specific than S_2. A subject s in S_1 has the properties that the subjects in S_2 has. Moreover some other properties exist to make s classified into S_1. For example, Hao is a member of group $Coder$ because he is assigned to some coding task of the projects in KnowDive research team. If these coding tasks are not assigned to Hao, he might be classified only in group $KnowDive$ as he has the identity of an exchange student in the team. The partial order between subjects can grasp more than static subject properties. In a pervasive environment, all the dynamic subject properties such as position, number of access attempt, number of connections, access purpose, etc., can be used to classify subjects. For example, a subject is classified into the group $ConnectionLessThan2$ for the moment and dropped to the group $NoConnection$ for the next moment. $O_1 \geq O_2$ means O_1 is an object class more specific than class O_2 similar to the subject.

$P_1 \geq P_2$ means the pair $(s,\ o) \in P_1$ has more specific properties than those pairs in P_2. These properties do not necessarily relate to s or o but rather concerning the permission itself. For example, the operation to update implies the operation to read and write can be represented as as axioms $Update \geq Read$ and $Update \geq Write$. In a pervasive environment, actual access time could be an important property of permissions. For example in Fig. 4b, *access on weekdays* and *access on weekends* should be treated differently.

A general intuition is that the more powerful, the smaller group; the more powerful, the bigger class; the more powerful, the more specific permission with less $(u,\ o)$ pairs.

4.3 Rule Representation

A `PERMISSION` in *RelBAC* is a binary relation which can be represented in *RelBAC* logics as a DL role. With the default DL constructors *full existential quantifier, value restriction*, a `RULE` in *RelBAC* may have the following forms:

$$S \sqsubseteq \exists P.O \tag{1}$$
$$S \sqsubseteq \forall P.O \tag{2}$$
$$S \sqsubseteq \geq nP.O \tag{3}$$
$$S \sqsubseteq \leq nP.O \tag{4}$$
$$S \sqsubseteq \forall O.P \tag{5}$$
$$O \sqsubseteq \exists P^{-}.S \tag{6}$$
$$O \sqsubseteq \forall P^{-}.S \tag{7}$$
$$O \sqsubseteq \geq nP^{-}.S \tag{8}$$
$$O \sqsubseteq \leq nP^{-}.S \tag{9}$$
$$O \sqsubseteq \forall S.P^{-} \tag{10}$$

Different access control rules can be designed with the same (S, P, O) tuple thanks to the formulas above. Formula (1) assigns to each member in S the permission P to some objects in O without specifying which concrete object should be accessed. For example, $Coder \sqsubseteq \exists Write.Code$ states that a coder can access (*write*) some code, but not necessarily Code1.0, maybe some other code files. This rule is mainly used for general access description, and specific access control will have to be specified later to clarify the meaning. A further rule such as $Write(Rui, Code1.0)$ may be added to the rule base. On the other hand, Formula (2) assign permission on specific object class to subject such that for subject in S the range of P is only O. For example, $Coder \sqsubseteq \forall Write.Code$ states the *write* permission for group $Coder$ means to write $Code$ file only, nothing else can be written except those in $Code$.

Besides, general rule can express cardinality permission assignment with help of *number restriction* constructors. In addition to rule form (1) and (2), we can express cardinality related rule as (3) and (4). Moreover, with *conjunction, disjunction* and *negation* constructor, we can express *exact cardinality* such as 'coders can write exactly 2 code files' in rule $S \sqsubseteq \neg \geq n+1\ P.O \sqcap \geq n\ P.O$ or *strict cardinality* such as 'coders can write more than 2 code files' $S \sqsubseteq \geq nP.O \sqcup \neg \leq nP.O$. As a special case, when $n = 1$, we have *function* relation with one-to-one mapping.

We discussed rule representation in *RelBAC* starting from a subject group, while rules start with object class follow a similar intuition with inverse of permissions with Formulas (6–10). They are not discussed in details, but equally important as subject starting rules because they offer a way to administrate access control where object information are clearly studied and emphasized such as in a GRID computing environment, a rule may state that only those computers with system load under 40% can be accessed.

RelBAC can enforce rules about individuals thanks to the *set* constructor mentioned in Sect. 4.1 as $\{a_i\}$. For example, $\{hao\} \sqsubseteq Coder$ states that Hao is a member of the group Coder; $Coder \sqcap \neg\{hao\} \sqsubseteq \exists Update.(Alpha \sqcup \{code1.0\})$ states that all the coders except Hao can update some Alpha code plus Code 1.0.

5 Using *RelBac* on Access Control in Pervasive Environment

Here is a case study of an access control scenario under pervasive environment: a project meeting event. Practical issues such as the evolving environment and scalability are also addressed in this section.

Suppose there will be a meeting for project named 'OpenKnowledge' in Room 7 of Faculty of Science on Aug. 28th from 16:30 till 17:30, 2008. Each attendant (local or outer)

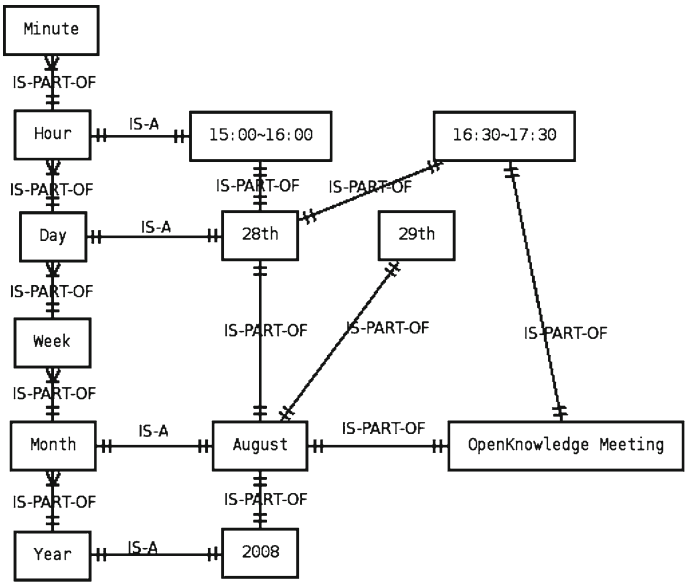


Fig. 5 Time Hierarchy. Year, Month, etc., sets are predefined and new entity sets such as August of 2008 can be extended to the model to capture the contexts. Here we use the crow's foot notation of ER diagram for simplification

of the meeting should be the partner of the project. There exists a monitor system of digital cameras that can recognize the people who join the meeting and also an IP analyzer to associate digital device location to the device user. All the attendants should be able to access the content file and demo scripts while in the room during the meeting. The demo will be accessible for a week afterwards for demonstration purpose. People may leave the room without mentioning the AC admin. The meeting could be extended for some time rather than strictly ends at 17:30.

In this scenario, the subjects of the access control could be anonymous (outer) or identifiable (local). On entering the meeting room, a person might be identified by the camera through face recognition and the location of the person is reported to the pervasive information center. When the wireless device such as a PDA or a laptop is in use, the URI of the device could be identified by the information center and evaluated with the predefined association to the owner of the device. The access control system may enforce that Rui's laptop connection will be valid only if Rui is located together with it. In the view of the sample user Rui, he can import part of the social network in Fig. 2, i.e., the branch of 'KnowDive' to a new user classification specially for the meeting. He may refer to the hierarchies shown in the Figs. 5 and 6 to classify the subjects with the dynamic information of the contexts provided by the environment. Here in Fig. 5, the entity in each concept are not individual time events, but the subjects, objects and even permissions at the corresponding events. For instance, the entity set named '16:30–17:30' refers to all the subjects and objects in that time event. These entities should be further filtered by conditions such as type (*Subject* or *Object*) and location information in Fig. 6. Thus a meeting partner in the case will be a member of the group

$$KnowDive \sqcup (OpenKnowledge\,Meeting \sqcap Room7)$$

Fig. 6 Position Hierarchy

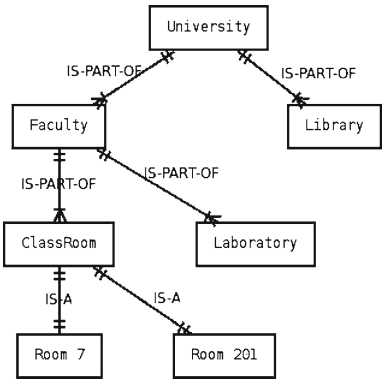
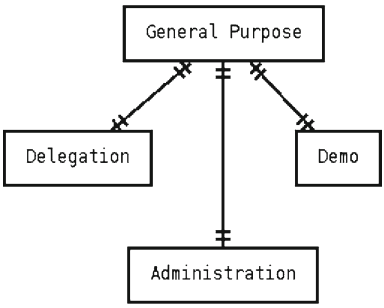


Fig. 7 Purpose Hierarchy



in which $KnowDive$ specifies those attendants from the local group and the second part of the concept represent those appear in Room 7 during the meeting which includes all the outer partners who join the meeting. Here in Figs. 5 and 6, 'IS-PART-OF' relation is used as an extension of *RelBAC* to represent precise relations between components of groups or classes. And it can be formalized with subsumption axiom in *RelBAC* logic as 'IS-A' relation. Similarly, permission *Read* can be specially constraint to *Read During the OpenKnowledge Meeting* or a week afterwards with

$$Read \sqcap (OpenKnowledge\,Meeting \sqcup Aug29\text{–}Sep4)$$

where $Aug29\text{–}Sep4$ is the concept for anything during the time event from August 29th till September 4th.

With the user group and permission represented above, *RelBAC* can enforce with the following rule that *attendants of the meeting can read during the meeting project content files including demos and demos are still available for the coming week after the meeting with demonstration purpose*.

$$KnowDive \sqcup (OpenKnowledge\,Meeting \sqcap Room7) \sqsubseteq$$
$$\forall((ProjectContent \sqcup Demo) \sqcap Room7).(Read \sqcap OpenKnowledge\,Meeting)$$
$$\sqcup \forall Demo.(Read \sqcap Aug29\text{–}Sep4 \sqcap Demonstration)$$

The context-aware environment helps to collect real-time information such as the camera recognize an attendant's leaving the room so that she is revoked from the group of 'Room 7'. If she is not a local 'KnowDive' member, she will be deprived the access to the project contents

during this absence. Neither her laptop identified by IP recognition technologies in Room 7 can access the contents anymore. This will prohibit a mis-use of others' devices. Another instance is that the meeting is delayed for 15 min because of some VIP's late and will not end till 18:00 because of some heat discussion. The content files should be still available for the attendants even after the scheduled ending time 17:30 because we may have a rule as follows.

$$16:30\text{–}17:30 \sqsubseteq \textit{OpenKnowledge Meeting}$$

6 Conclusions and Future Work

The access control problem for context-aware computing becomes serious because of the vast amount of various data to be protected. In this paper, we proposed a new model in ER diagram, *RelBAC* (Relation-Based Access Control) for context-aware environment. A Description Logic based access control domain specific logical framework is also proposed to formalize and reason about the model. *RelBAC* offers a compact formal language and rich expressiveness in arity related access control. Automated reasoning can be performed on the knowledge base formed by *RelBAC* formulas.

RelBAC has been implemented with subject and object hierarchy into light-weight ontologies [8]. Preliminary evaluation shows challenges and potentials of automated reasoning. The future work of this research could be the optimization of general purpose DL reasoners for *RelBAC* and the implementation of *RelBAC* in more challenging fields. Automated or semi-automated tools can be developed via the automated reasoning through state of the art DL reasoners.

References

1. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The description logic handbook: Theory, implementation, and applications*. New York: Cambridge University Press.
2. Bouquet, P., Giunchiglia, F., Harmelen, F. V., Serafini, L., & Stuckenschmidt, H. (2003). C-owl: Contextualizing ontologies. In: *Journal Of Web Semantics*, Springer, pp. 164–179.
3. Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., & Abowd, G. D. (2001). Securing context-aware applications using environment roles. In: *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies* (pp. 10–20). New York, NY: ACM. http://doi.acm.org/10.1145/373256.373258
4. Damiani, M. L., Bertino, E., Catania, B., & Perlasca, P. (2007). Geo-rbac: A spatially aware rbac. *ACM Transactions on Information and System Security, 10*(1).
5. Damianou, N., Dulay, N., Lupu, E. C., & Sloman, M. (2000). *Ponder: A language for specifying security and management policies for distributed systems*. Imperial College Research Report DoC 2000/1. URL:citeseer.ist.psu.edu/damianou00ponder.html.
6. Emami, S. S., Amini, M., & Zokaei, S. (2007). A context-aware access control model for pervasive computing environments. *Intelligent Pervasive Computing, 0*, 51–56. http://doi.ieeecomputersociety.org/10.1109/IPC.2007.6
7. Ferraiolo, D. F., Sandhu, R. S., Gavrila, S. I., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *Information and System Security, 4*(3), 224–274. URL:citeseer.ist.psu.edu/ferraiolo01proposed.html.
8. Giunchiglia, F., Marchese, M., & Zaihrayeu, I. (2007). Encoding classifications into lightweight ontologies. *Journal of Data Semantics, 8*.
9. Giunchiglia, F., Zhang, R., & Crispo, B. (2008). Relbac: Relation based access control. In: *SKG '08: Proceedings of the 2008 Fourth International Conference on Semantics, Knowledge and Grid* (pp. 3–11). Washington, DC: IEEE Computer Society. http://dx.doi.org/10.1109/SKG.2008.76.
10. Hulsebosch, R. J., Salden, A. H., Bargh, M. S., Ebben, P. W. G., & Reitsma, J. (2005). Context sensitive access control. In: *SACMAT '05: Proceedings of the tenth ACM symposium on Access control*

models and technologies (pp. 111–119). New York, NY: ACM. http://doi.acm.org/10.1145/1063979.1064000.
11. Joshi, J., Bertino, E., Latif, U., & Ghafoor, A. (2005). A generalized temporal role-based access control model. *IEEE Transactions on Knowledge and Data Engineering, 17*(1), 4–23.
12. Kagal, L. (2002). *Rei : A Policy Language for the Me-Centric Project*. Tech. rep., HP Labs. http://www.hpl.hp.com/techreports/2002/HPL-2002-270.html.
13. Kagal, L., Finin, T., & Joshi, A. (2003). A policy language for a pervasive computing environment. In: *POLICY '03: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks* (p. 63). Washington, DC: IEEE Computer Society.
14. Kulkarni, D., & Tripathi, A. (2008). Context-aware role-based access control in pervasive computing systems. In: *SACMAT*, pp. 113–122.
15. Lampson, B. (1971). Protection. In: *Proceedings of 5th Princeton Conference on Information Sciences and Systems, Princeton, 1971*. Reprinted in ACM Operating Systems Rev. 8, 1, pp. 18–24
16. Mccarthy, J. (1993). Notes on formalizing context. In: *Proceedings of the Thirteenth International Joint conference on Artificial Intelligence* (pp. 555–560). Morgan Kaufmann.
17. Moyer, M. J., & Ahamad, M. (2001). Generalized role-based access control. In: *ICDCS*, pp. 391–398.
18. Uszok, A., Bradshaw, J. M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., & Aitken, S. (2004). Kaos policy management for semantic web services. *IEEE Intelligent Systems 19*(4), 32–41. http://doi.ieeecomputersociety.org/10.1109/MIS.2004.31.
19. Zhang, G., & Parashar, M. (2004). Context-aware dynamic access control for pervasive computing. http://citeseer.ist.psu.edu/687356.html.
20. Zhang, R., Artale, A., Giunchiglia, F., & Crispo, B. (2009). Using description logics in relation based access control. Tech. rep., University of Trento. http://eprints.biblio.unitn.it/archive/00001611/01/024.pdf.

Author Biographies

Rui Zhang got his PhD on Informatics in University of Trento, Italy in 2009. His research interests include automated reasoning, access control and logical modeling. Now he is a junior lecturer in the faculty of Computer Science and Technology in Jilin University, China. He has published four papers on International Conferences. His PhD thesis has been accepted as a book in the series of Studies on the Semantic Web. He is also the committee member of the Ontology Matching Workshop in ISWC09.

Fausto Giunchiglia is a full professor in the Department of Information Engineering and Computer Science (DISI), University of Trento and also a Honorary Professor at Jilin University, Changchun, China, since April 21, 2008. His research interests are broad, such as Artificial Intelligence, Formal Methods, automated reasoning, theorem proving, contexts and contextual reasoning (and logics for modeling it), abstract reasoning, meta-theoretic reasoning, etc., and has published more than 230 papers on International conferences and journals. He has been the editorial board of many books and journals.

Bruno Crispo is Associate Professor at the University of Trento. He received his Ph.D. in Computer Science from the University of Cambridge, UK, in 1999. His main research interests include security and privacy in large distributed systems, RFID and sensors security, and security protocols and access control. He has published a book and more than 80 research papers on these topics in technical journals and international conferences.

Lingyang Song received PhD in differential space time codes and MIMO from the University of York, UK, in 2007. He worked as a postdoctoral research fellow at the University of Oslo, Norway, until rejoining Philips Research UK in March 2008. Now, he is with School of Electronics Engineering and Computer Science, Peking University, China. He is co-inventor of a number of patents and author or co-author of over 60 journal and conference papers. Currently he is the Editorial Board of many International journals, conferences and workshops.