Capturing Semantics for Information Security and Privacy Assurance

Mohammad M.R. Chowdhury[1], Javier Chamizo[2], Josef Noll[1], and Juan Miguel Gómez[2]

[1] UniK-University Graduate Center
Post Box 70, N-2027 Kjeller, Norway
{mohammad, josef}@unik.no@unik.no
[2] Escuela Politécnica Superior
Universidad Carlos III de Madrid
Avda. de la Universidad 30, Leganés, Madrid, Spain
chaminet@gmail.com, juanmiguel.gomez@uc3m.es

Abstract. Security and privacy assurance is indispensable for ubiquitous access to information and resources. This paper focuses on the security and privacy provisions in a restricted organizational environment through access control mechanism. It includes the representation of the semantics of an organization and its access control mechanism exploiting the Web Ontology Language. The system controls access to the resources of an organization through differential access privileges. These are formulated based on the roles of the individuals, and the projects and departments they belong to. Instead of explicit definitions, some additional facts of the mechanism are inferred by executing semantic rules using the Jess rule engine over the designed ontology. These information are then passed back to the ontology to enrich it. The ontology is designed to cope with the organization restructuring with minimal efforts.

1 Introduction

Ubiquitous computing and connectivity together with extensive diffusion of portable devices allow users to access information/resources/services anytime and anywhere even when they are on the move. However, these access scenarios demand security and privacy assurance which is not a trivial job in today's increasingly connected but dynamic systems. In this regard, Professor Dr. Eugene Spafford said [1],

> The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts.

This paper focuses on the security and privacy provisions in a restricted organizational environment through access control mechanisms. Access control in distributed and dynamic systems is crucial for secure service access. It is included

F.E. Sandnes et al. (Eds.): UIC 2008, LNCS 5061, pp. 187–200, 2008.

as one of the main sections in ISO/IEC 27000 series[1], an information security standard published by the ISO and the IEC. We believe that the capabilities of semantic technologies can contribute to mitigate these problems. The impact of Semantic Web technology is wide ranging. The Project10X (a consulting firm)[2] study found that more than 190 companies including Adobe, Google, HP, Oracle and Sony are involved in developing Semantic Web based tools. But making it easier to comb through online data carries security implications. Among the challenges of security issues, policy-awareness and access control to Web resources play a major role, particularly given that these are two of the most significant requirements of information access and exchange.

Design and maintenance of access control constraints in organizations are challenging problems as company structure, roles, user pools, security requirements are always changing. Conceptual organizational semantics and its access control mechanisms are fomally represented in an ontology using Web Ontology Language. The company system controls access to the resources of the organization by providing differential access privileges. In this ontology, these privileges are formulated based on the roles of the individuals. In addition, it considers which projects or departments they belong to. The proposed solution is designed to cope with the dynamic nature of an organization. This work is an extension of our previous work [2], where the concepts were so static that it could not reflect the organizational changes. This paper deals with a complex situation where an employee plays multiple roles across different departments and projects.

The paper is organized as follows. The next section discusses the problem statements and our use case scenario. Section 3 briefly describes the Semantic Web technologies. The ontology representing organizational semantics are described in section 4. In Section 5, we illustrate the access control mechanism, the processes and results of inference based on the proposed ontology. In the next section, proposed solution is evaluated in the context of organizational restructuring. Section 7 contains overview of the related works and the paper concludes summarizing the paper and briefly stating the future works.

2 Problem Statement: Information Security and Privacy Assurance

Nowadays people in business organizations increasingly work in project oriented environments. Some of the projects deal with company sensitive information which should not be leaked to unauthorized employees. The project members come from different departments. They don't enjoy the same rights or privileges within a project environment. This is more prevalent while accessing resources owned by the departments or projects. There are situations where a person

[1] ISO 27002 - The Information Security Standard, http://www.standardsdirect.org/iso17799.htm [accessed on Jan. 4, 2008]

[2] The Project10X Special Report, http://www.semantic conference.com/semanticwave.html

holds multiple roles and privileges. Employees with different levels of privileges are expected to access resources through the Intranet or Internet.

Fig. 1 illustrates the use case scenario which describes a specific organizational environment. It deals with the roles like employee (department in general), supervisor, project leader (PL) and project member (PM). Telenor R&I (Department A) and its planning department (Department B) both involve in project release 7 and 8. Release 9 only resides in planning department. Each of the departments and projects has its own resources. A person has multiple roles like Josef Noll is not only a supervisor (Department A) and project leader (Release 9) but also a project member (Release 7). He should have access to corresponding department and project resources where he involves in. So, access rights depend on one's role into the respective departments and projects. Following are some of the examples of restricted access scenarios,

1. Supervisor is the head of a department. Department owns some resources (administrative resources, documents, deliverables). He can read and write the documents. He can edit its administrative resources and give final approval to the deliverables. Supervisor can also monitor status of department's employees who work in different projects.
2. Department's employees will have only read and write privileges to its documents.
3. Departments participate in different projects. Project leader leads a project. He can read and write project's documents. He can edit its administrative resources and give final approval to the project deliverables.
4. Besides leader, projects have members. They can only read and write project's documents.

Therefore, the architecture manages access to resources not only based on the roles but also based on the involvement in organizational divisions (departments, projects). Access scenarios of the use case are described in table 1.

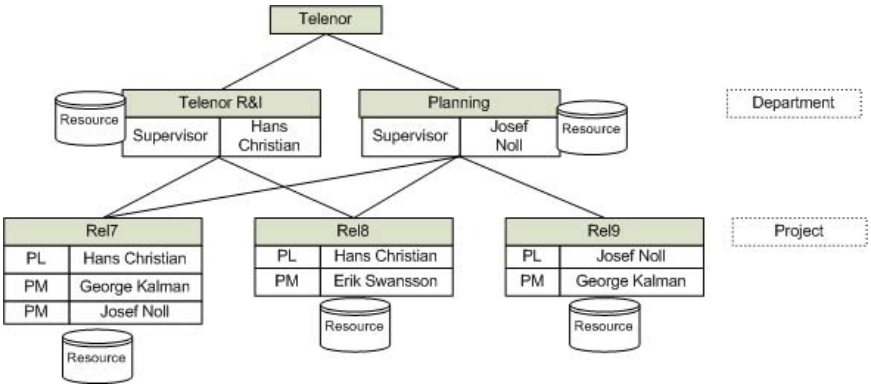

Fig. 1. The use case scenario

Table 1. Roles and privileges to access corresponding resources

Employee	Role	Privilege	Access to Resources
Josef Noll	Supervisor	Administrator	Admin. Dept.A
		Final Approval	Deliverables Dept.A
		Read Write	Documents Dept.A
	Project Leader	Administrator	Admin. Rel9
		Final Approval	Deliverables Rel9
		Read Write	Document Rel9
	Project Member	Read Write	Documents Rel7
Hans Christian	Supervisor	Administrator	Admin. Dept.B
		Final Approval	Deliverables Dept.B
		Read Write	Documents Dept.B
	Project Leader	Administrator	Admin. Rel7&Rel8
		Final Approval	Deliverables Rel7 &Rel8
		Read Write	Documents Rel7 &Rel8
George Kalman	Employee	Read Write	Documents Dept. A
	Project Member	Read Write	Documents Rel8
			Documents Rel9
Erik Swansson	Employee	Read Write	Documents Dept. A
	Project Member	Read Write	Documents Rel8

3 Representation of Organizational Semantics

We advocate that access control solutions should adopt semantic technologies as the key building blocks for supporting expressive representation of organizational semantics and reasoning. This section briefly describes the technologies and clarifies our claims.

3.1 Using Ontology

A common format of information and data representation will likely never be achieved. Efficient management of information and data is possible by capturing the common and understandable meaning of them formally and unambiguously [16]. Ontologies [15] are the cornerstone technology of Semantic Web, providing structured vocabularies that describe a formal specification of a shared conceptualization. It is used to capture the knowledge about a domain of interest in the form of concepts and their relationships. It permits the description of organizational structures, roles, privileges and resources at different levels of abstraction and support reasoning about both the structure and the properties of the elements that constitute the system. We believe that designing a consistent ontology based on a sound conceptual foundation is worthwhile because it can be reused in various organizations to control access to their resources.

3.2 Introduction to the Web Ontology Language

Among the different ontology languages, we are focusing on the Web Ontology Language (OWL[3]) suggested by the World Wide Web Consortium (W3C). It is a markup language that builds on RDF[4] and RDF Schema. OWL is chosen because it provides more vocabularies for describing concepts and properties (e.g. relations between concepts, cardinality, equality, richer typing of properties, etc) than that supported by XML, RDF, and RDFS. There are three species of OWL: OWL Lite, OWL DL and OWL Full and these are designed to be layered according to their increasing expressiveness.

3.3 Description Logics for OWL

Between the three different sub-languages OWL offers, we decided to use OWL DL. It is based on Description Logics (hence the suffix DL). These are the decidable part of First Order Logic[5] and are therefore amenable to automated reasoning. Though OWL DL lacks in expressivity power compared with OWL Full, it maintains decidability[6] and regains computational efficiency. The computational efficiency is an important feature since it is expected to support scores of relations. The mechanism is supposed to evaluate and grant permissions to access resources, it seems necessary to add reasoning support with it. In order to achieve more expressivity and decidability, we use Semantic Web Rule Language (section 5.1) which is designed as an extension of OWL DL.

4 Descriptions of Organizational Semantics

We assumed that company employees are already authenticated to the system through some secure means. Ontology models the organizational structures described in section 2.

4.1 Defining Concepts through Classes

Fig. 2 illustrates the proposed ontology in conceptual level. OWL classes are the concrete representations of concepts. In the proposed ontology, *Identity* class defines the identities of the company employees. We specify *Company*, *Department* and *Project* as subclasses of *Work Unit* in order to avoid defining explicit relationships between department/project and roles. We follow the set theory (eq. 1) while defining the class hierarchy of *Role* considering the fact that supervisor of

[3] OWL Overview: `http://www.w3.org/TR/owl-features/`

[4] RDF builds on URI and XML technologies. The specifications provide a lightweight ontology system.

[5] First Order Logic (FOL), http://en.wikipedia.org/wiki/First-order_logic

[6] Logics are decidable if computations/algorithms based on the logic will terminate in a finite time.

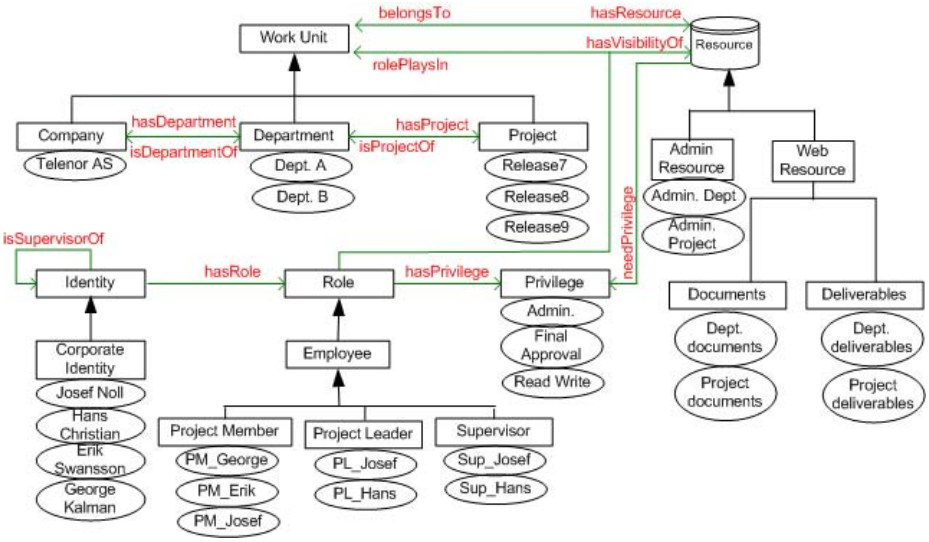

Fig. 2. The ontology: class, property, instance

a department is also an employee of it. The same is true for project leader and member. Class hierarchy is a critical issue in inheritance of properties.

$$\{Supervisor, ProjectLeader, ProjectMember\} \subseteq Employee \subseteq Role \quad (1)$$

We divide the resources into subclasses, administrator resources and web resources. Web resources are further divided into documents and deliverables. These resources are related to appropriate privileges. Privileges are designed in accordance with the individual's roles in the organization. In this paper, role is the positional hierarchy of employees. Through this, individuals are restricted to access correct resources. As for example, administrative resources are related to the *Admin_Privilege* to ensure that only the roles having administrative privilege can access these.

4.2 Realizing Concepts through Instances

OWL classes are interpreted as sets that contain instances or individuals. Fig. 2 illustrates the instances of classes in ellipses. Instances of the identities are defined here simply as their names. *Admin*, *Final approval* and *Read write* instances of privilege are added. But new instances can be added (section 6) whenever necessary. Instances of the resources are added which correspond to the resources owned by the departments or projects. Instances to the subclasses of *Role* are added in accordance with the individual roles to realize multiple roles of a person. As for example, *Sup_Josef* instance of *Supervisor* corresponds to the supervisor role of Josef Noll. Similarly, *PM_Josef* corresponds to the project member role of Josef Noll.

4.3 Defining Relations through Properties

Properties are the binary relations between the two things, more specifically between the instances of classes. A property relates instances from the *domain* with the instances from the *range*. Syntactically, *domain* links a property to a class and *range* links a property to either a class or a data range. Due to the class hierarchy (eq. 1) and *domain* and *range* specifications, subclasses inherit the relationships between the respective classes. Fig. 2 also provides the properties and their relationships with classes. *rolePlaysIn* property specifies the fact that in which specific *Work Unit* (departments/projects) a *Role* instance plays its role. Ontology is supposed to answer who can see/access which resources and *hasVisibilityOf* property defines this situation. *isSupervisorOf* explains who is the supervisor of whom in a department. The relationships of *hasVisibilityOf*, *isSupervisorOf* are not defined explicitly. These relationships of the properties are filled in through the inference process.

5 Access Control by Enhancing the Expressivity of OWL

Access control is achieved by enhancing the expressivity of OWL through the inference process. This section describes the access control logic and expressivity needs, making a review of the language used and its benefit to the goals pursued.

5.1 Introduction to the Semantic Web Rule Language

The expressivity provided by the OWL is limited by tree like structures [17]. This means that knowledge cannot be inferred from indirect relations between the entities, however the solution spends most part of his power in inferring indirect relationships that will determine whether a subject has access to a resource or not and which are its privileges over it. Hierarchical structures as defined before and inherent relationships between working units and hierarchies of resources are a perfect field where inference can extract these knowledge. We did inference through the rule support over the ontology. and used Semantic Web Rule Language (SWRL[7]) which pretends to be a complimentary feature of OWL. SWRL is roughly the union of Horn Logic and OWL. As any Horn Logic based language, rules are defined as a set of precedent and consequent states.

5.2 Inference Results

Objects of the properties, *hasVisibilityOf* and *isSupervisorOf* are filled in through the inferred knowledge from executing the rules. We use Jess rule engine to run the rules. First, OWL ontology and SWRL rules are transferred to Jess. Running the engine then initiates the inference process, generates knowledge as Jess facts. This inferred knowledge can be used by the external interface or can optionally be passed back to the ontology to enrich it. All these actions are user-driven. Rules are formulated using the SWRL as follows,

[7] The Semantic Web Rule Language, http://www.w3.org/Submission/SWRL/

- Rule1: Over which resource a Role has Visibility/Access?
  $Employee(?Em) \wedge rolePlaysIn(?Em, ?X) \wedge hasPrivilege(?Em, ?Y) \wedge belongsTo(?Z, ?X) \wedge needPrivilege(?Z, ?Y) \longrightarrow hasVisibilityOf(?Em, ?Z)$

- Rule2: Who is supervisor of whom?
  $Dept\_Employee(?DepEm) \wedge hasRole(?Y, ?DepEm) \wedge Department(?Dep) \wedge rolePlaysIn(?DepEm, ?Dep) \wedge Corporate\_Identity(?ID) \wedge Supervisor(?Sup) \wedge hasRole(?ID, ?Sup) \wedge rolePlaysIn(?Sup, ?Dep) \longrightarrow isSupervisorOf(?ID, ?Y)$

Fig. 3 and 4 illustrates the inference results of rules execution. All the relationships in the ontology have not been explicitly defined. As for example, *hasVisibilityOf* relationship of project leader Hans (*PL_Hans*) has not been defined (circled in fig. 3). The figure displays that 25 relationships are inferred, which answers the resources over which the roles have the required visibility/access (Rule 1). Our investigation shows that the knowledge are inferred as expected. The inference results are exported back to the ontology to fill these empty relationships (fig. 3 shows that 25 relationships are transferred back to the OWL knowledge). From Tab. 1, it is evident that Josef Noll is the supervisor of George Kalman, which was not explicitly defined. Execution of Rule 2 shows the similar result (Fig. 4). This can be used in a special situation, when supervisor wants to check the status of an employee of his department to a project where he is not involved.

6 Evaluation

The proposed solution is more maintainable since there exist a general schema for any organization that can easily be adapted, thanks to its expressivity capacity nearer to human understanding. Expressivity and inference capacities avoid the inclusion of redundant information in the ontology.

The proposed ontology can reflect the organization changes of a company with minimum efforts. Now, we are going to describe a situation of this. A new department, Audit has been created in the company. Roman Stanek and Peter Johansson has joined as the supervisor and employee (auditor) of the department respectively. The department has budgets, and Peter audits the budgets. Audit department prepares the audit reports of company.

It is expected that auditor (Peter) as well as supervisors of the departments can only check department's budget. Roman and Peter only have access to the company audit reports that belong to the Audit department only. As a supervisor of Audit department, Roman can give final approval to these. The corresponding actions to reflect these changes into the ontology are described in the following points. Through this, we are going to evaluate the strength of the proposed ontology.

- Add new identity instances: *Roman Stanek* and *Peter Johansson*.
- Add new department instance: *Audit*

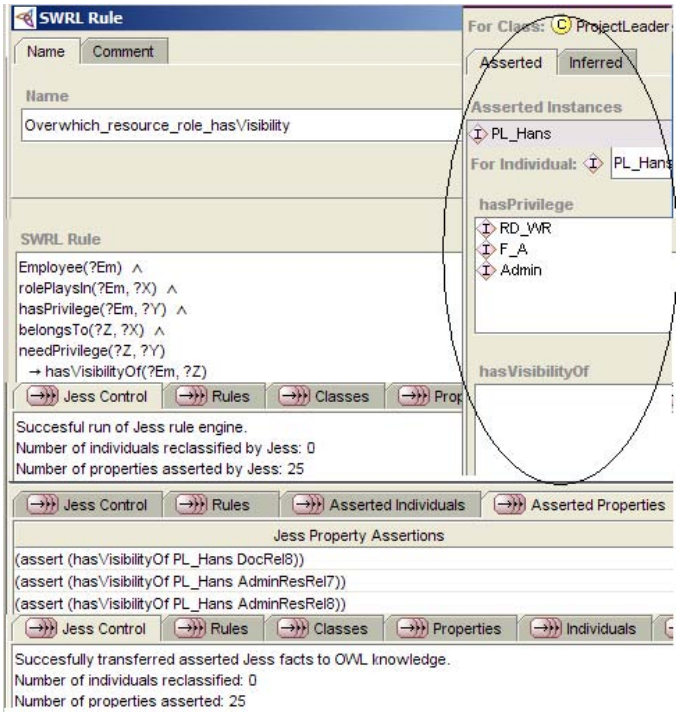

Fig. 3. Inferred results executing Rule 1

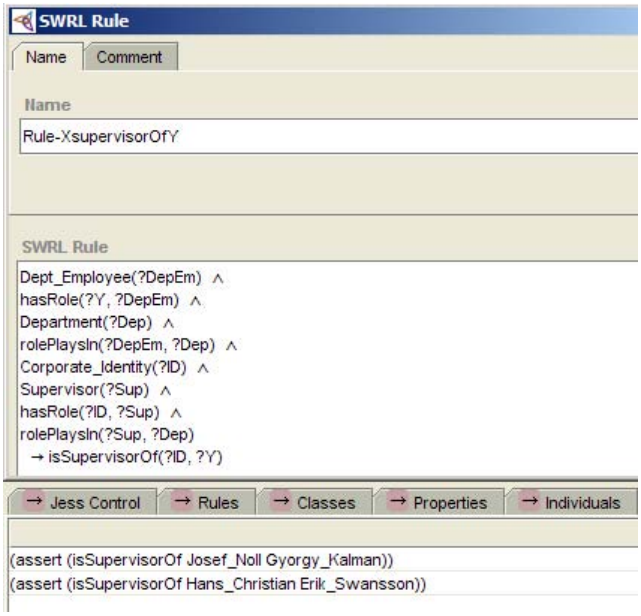

Fig. 4. Inferred results executing Rule 2

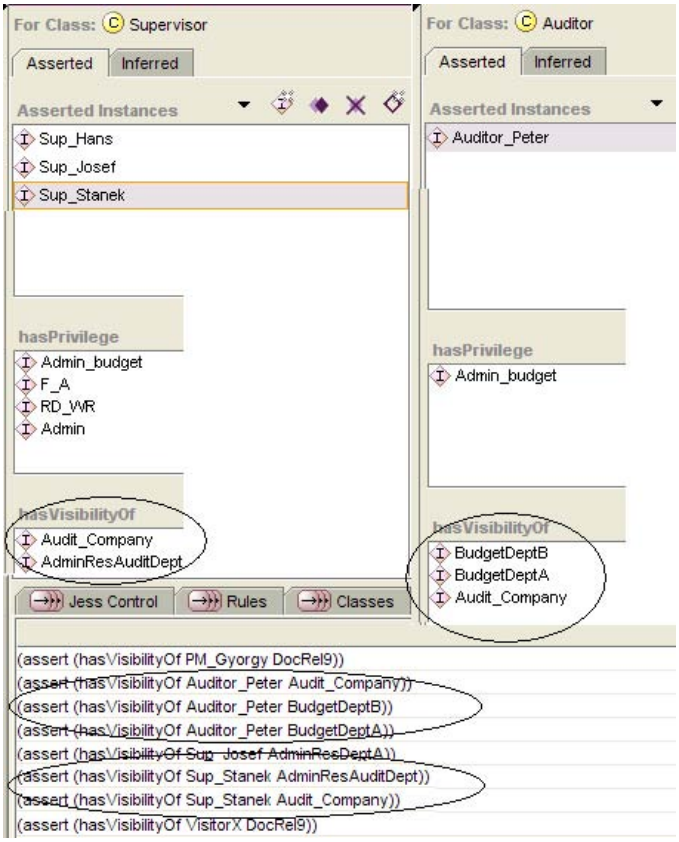

Fig. 5. Inferred relationships when Rule 1 is executed

Fig. 6. Inferred relationships when Rule 2 is executed

- Add new subclass of *Resource*: *Budget & Audit*. It contains three instances: audit report of company (*Audit_Company*), *BudgetDept.A* and *BudgetDept.B*. A new subclass of resource is created because it needs a different privilege to support its privacy requirements. Besides, the department contains an administration resource (*AdminResAuditDept*).
- Add new subclass: *Auditor* within the class tree: Role-CompanyEmployee. Add *Auditor_Peter* instance of it.
- Add new instance of *Supervisor* role: *Sup_Stanek*.

- Add a privilege instance *Admin_Budget* to ensure only auditor and supervisor have access to budget. This instance is related to *Auditor_Peter* and *Budget Dept.A* & *Budget Dept.B*.
- Fill up all the corresponding relationships.
- If we execute Rule 1, new relationships are inferred with *hasVisibilityOf* property. These are exported back to the ontology to fill in (circled in the figure) the empty relationships. Fig. 5 shows these new relationships for instance of auditor role *Auditor_Peter*. As expected, Peter can only have access to company audit reports and Budgets of Dept.A and B. Similarly, fig. 6 shows the fact that Roman Stanek is the supervisor of Peter Johansson.

It requires only few changes in the ontology. Among these changes, only two new subclasses have to be created. Otherwise, all the remaining additions are in the instances which is quite apparent. Conceptually, proposed ontology can follow the organizational changes. One of the ways of simplifying access rights management is to store the access control matrix using access control list (ACL). Though ACLs are simple to implement, its management steps are quite tedious especially when it is required to revoke somebody's privilege. The proposed ontology provides a simple mechanism to revoke somebody's role or privilege. One can simply delete the relationship between the role instance and the privilege instance to withdraw the privilege and afterward can delete the corresponding role instance to repeal the subject's role entirely. Among the few disadvantages, it is computationally expensive and decidability not guaranteed by SWRL. In addition, XML processing is slower than that of database. But ontologies can be stored in the relational databases or even be mapped from them.

7 Related Works

The significance of adding privacy-enhancing technologies (PET) in virtual community networks is overwhelming [5], [6]. Information security and privacy requirements are more evident in a business environment. These issues are handled in this paper through access control mechanisms in the context of project oriented corporate working environment.

We considered the concept of Role Based Access Control (RBAC) as a part of our access control mechanism. Sandhu introduced RBAC in [4] with early multiuser computer systems, where users' access permissions are associated with roles, and users are made members of appropriate roles. A major advantage of RBAC is its ability to constraint access based on the concept of separation of duties which significantly simplifies the management of permissions. In the proposed solution, access control also depends on how the organizational structures.

There are two types of RBAC constraints: dynamic and static. Authors in [3] described an approach of RBAC with dynamic constraints through an automated reasoning technique. Though we focused on static constraints on roles, rules were included within the ontology to infer new knowledge which can be passed back to the ontology. Through this verification of access control constraints defined

in the ontology are also achieved. The proposed solution can also adapt to ever changing company structure with less effort.

Ubiquitous connectivity not only permits users to access resources anytime and anywhere but also complicating its control due to user/device mobility. To integrate this pervasive behavior, a context-aware access control framework has been developed for secure collaborations in pervasive computing environments [18]. It followed two main design principles: context-awareness to control resource access and to enable dynamic adaptation of policies based on context, and the use of OWL DL for context/policy specification. The authors considered location and time of the meeting as the context information collected dynamically. We used static contexts like, which departments/projects someone is involved though these were predefined.

Among the access control technologies, ACL is widely used. The semantics of ACLs have been proved to be insecure in many situations. Instead of maintaining a centralized ACL, a trust based group has been proposed to delegate access rights in [7], [8] where FOAF (friend of a friend) [9] based social networks acted as a mean for the delegation. A private key based signature scheme was proposed to ensure the privacy of networks and users. But it requires secure distribution and maintenance of keys. A similar concept of trust or reputation has also been used by [10] to create and access communities. Distributed trust management approach is considered as one of the main components to secure the Semantic Web [11]. They intended to provide access to community resources and privacy solutions only by means of trust/reputation management. But access to sensitive business resources based on trust mechanism does not provide adequate security in business contexts. In addition, trust is affected by various factors and therefore difficult to quantify.

In [12], authors presented an approach to reduce the inefficiencies of the management (coordination, verification and validation, and enforcement) of many role-based access control policies and mechanisms using OWL. They focused on the representation of XACML (eXtensible Access Control Markup Language) policies in DL. In [13], Kolovski introduces an algorithm for the translation of a subset of XACML into DL with the goal of offering relevant analysis services using an OWL DL reasoner, Pellet[14]. In this paper, we also formalize the organizational semantics, roles and access privileges using OWL DL. Finini in his paper [11] also proposed using OWL for constructing ontologies which define policies/privileges.

8 Conclusions

In this paper, we addressed the security and privacy challenges of project-based organizations through access control mechanism exploiting Semantic Web technologies. In this regard, we developed an ontology to represent the conceptual structure of an organization and the roles of individuals. In the ontology, all the relationships between the entities have not been defined explicitly. Semantic rules facilitated expressing these additional knowledge. Jess rule engine executed

these rules and new facts were transferred back to the ontology to enrich it and check its validity.

Apart from these, we evaluated the inference capabilities of the proposed solution by restructuring the organization. As the proposed solution is based on centralized architecture, the future architecture should consider the scalability issues of it especially for a system of big enterprise. Our ultimate goal is to integrate this solution with a web application which controls access to resources.

References

1. Spafford, E.H.: director of the Purdue Center for Education and Research in Information Assurance and Security, Selected Quotes [accessed on January 4, 2007], `http://homes.cerias.purdue.edu/~spaf/quotes.html`
2. Chowdhury, M.M.R., Noll, J., Gomez, J.M.: Enabling Access Control and Privacy through Ontology. In: 4th International Conference on Innovations in Information Technology (Innovations 2007), Dubai, UAE (2007)
3. Dury, A., Boroday, S., Petrenko, A., Lotz, V.: Formal Verification of Business Workflows and Role Based Access Control Systems. In: International Conference on Emerging Security Information, Systems and Technologies (SECUREWARE 2007), Valencia, Spain (2007)
4. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. IEEE Computer 29(2), 38–47 (1996)
5. Chewar, C.M., McCrickard, D.S., Carroll, J.M.: Persistent virtual identity in community networks: Impact to social capital value chains. Technical Report TR-03-01, Computer Science, Virginia Tech (2003)
6. Walters, G.J.: Privacy and Security: An Ethical Analysis. Computers and Society, 8–23 (2001)
7. Kruk, S.R., Grzonkowski, S., Gzella, A., Woroniecki, T., Choi, H.-C.: D-FOAF: Distributed Identity Management with Access Roghts Delegation. In: 1st Asian Semantic Web Conference, Beijing, China (2006)
8. Kruk, S.R., Gzella, A., Grzonkowski, S.: D-FOAF Distributed Identity Management based on Social Networks. In: Demo session of ESWC 2006 (2006)
9. FOAFRealm project, `http://www.foafrealm.org/`
10. Choi, H.-C., Kruk, S.R., Grzonkowski, S., Stankiewicz, K., Davis, B., Breslin, J.G.: Trust Models for Community-Aware Identity Management. Identity. In: Reference and the Web IRW 2006, WWW 2006 Workshop, Scotland, May 23 (2006)
11. Finin, T., Joshi, A.: Agents, Trust, and Information Access on the Semantic Web. ACM SIGMOD 31(4), 30–35 (2002), Special Issue: Special section on semantic web and data management
12. Smith, M.A., Schain, A.J., Clark, K.G., Griffey, A., Kolovski, V.: Mother, May I? In: OWL-based Policy Management at NASA European Semantic Web Conference 2007, ESWC 2007 (2007)
13. Kolovski, V., Hendler, J., Parsia, B.: Analyzing Web Access Control Policies. In: 16th International World Wide Web Conference, WWW 2007, Alberta, Canada, May 8-12 (2007)
14. Pellet, an OWL DL reasoner, `http://pellet.owldl.com/`

15. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, Heidelberg (2001)
16. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (May 2001)
17. Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with Rules. In: International Semantic Web Conference 2004, pp. 549–563. Springer, Heidelberg (2004)
18. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments. In: International Semantic Web Conference 2006. LNCS, pp. 473–486. Springer, Heidelberg (2006)