# ROLE AND LOCATION BASED ACCESS CONTROL USING SEMANTIC WEB TECHNOLOGIES

SANDRO CIRULLI

STUDENT NUMBER: 10091144

SUPERVISOR: PROFESSOR DAVID DUCE

MODULE: P00998 - MSC DISSERTATION


MSc in Computer Science

Department of Computing and Electronics
Oxford Brookes University

2 September 2011

Sandro Cirulli: *Role and Location Based Access Control Using Semantic Web Technologies,* MSc in Computer Science, Oxford Brookes University, Oxford, August 2011.

## ABSTRACT

Accessing resources on a restricted area of a website is a common event that users experience daily. This is often modelled using role based access control (RBAC) and implemented by means of form authentication - i.e. via a user name and a password. However, this approach is somehow limited as the use of passwords for authenticating users has often been criticised on security grounds. On the other hand, Semantic Web technologies recently introduced new authentication protocols such as FOAF+SSL that may allow the development of a more secure and distributed access control system. In addition, due to the widespread use of mobile technologies, there is a need to take into account the user's location when determining access rights and it is therefore crucial to integrate location based access control (LBAC).

This dissertation discusses, develops and critically evaluates two access control systems respectively based on form authentication and FOAF+SSL. Both systems are developed in the frame of a real business scenario using Semantic Web technologies. Location constraints are also included in one developed system in order to combine both role and location based conditions.

This project shows how FOAF+SSL can offer a more distributed and secure approach to authentication, how role and local based conditions can be integrated to provide finer granularity in the access control system, and how the use of Java servlet technology allows the enforcement of dynamic access control policies.

*Computers are useless. They can only give you answers.*

*— Pablo Picasso*

## ACKNOWLEDGMENTS

I would like to express my gratitude to all those who gave me the possibility to complete this dissertation. I am deeply indebted to my supervisor Professor David Duce for his support, availability and guidance as well as for asking challenging questions that triggered my curiosity and a great number of ideas. I also gratefully acknowledge Dr Brian Matthews for discussing the project at its early stages and Professor Rachel Harrison for precious hints on the elicitation of business requirements.

I own my deepest gratitude to Giovanna Ferrante from Nielsen who gave me the initial idea for starting this project and provided me with crucial information to complete it. I am also grateful to André Miede who developed and made freely available his beautiful template for LaTeX with which I have typeset this dissertation.

I am deeply indebted to the ECIF Scholarship that freed me from financial worries and motivated me to concentrate on my studies during this academic year.

Finally, a special thank you to my friend Ruth Porter who went through the boring task of proofreading this dissertation and supported me with her sympathy, joy and rooibos tea.

# CONTENTS

# LIST OF FIGURES



# LIST OF TABLES

## LISTINGS



## ACRONYMS

ACL Access Control List

API Application Programming Interface

CA Certificate Authority

FOAF Friend of a Friend

JDK Java Development Kit

JRE Java Runtime Environment

LBAC Location Based Access Control

OWL Web Ontology Language

RBAC Role Based Access Control

RDF Resource Description Framework

REST Representational State Transfer

SSL Security Sockets Layer

SWRL Semantic Web Rule Language

URI Uniform Resource Identifier

W3C World Wide Web Consortium

# 1 INTRODUCTION

Accessing resources on a restricted area of a website is a common event that users experience daily. The importance of an access control system is therefore clear as such a system allows the granting or denial of permissions on resources. Although several approaches to access control were developed for computer systems (for an overview see Ferraiolo et al. (2003)), it is generally held that Role Based Access Control (RBAC) is the most employed access control approach in business environments. RBAC can be defined as an association of permissions with roles so that individual users can be granted permissions to perform actions on a computer system according to their role in the organisation (Power et al., 2009). Another approach to access control which has been introduced recently due to the widespread use of mobile devices is Location Based Access Control (LBAC). This takes into account the location of the user when granting access and allows a refinement of the granularity in the set of permissions.

With regard to web-based applications, RBAC is commonly implemented by using a combination of user name and password. Although such an implementation is extremely common on today's web applications, it is highly debatable whether this is the most secure and effective approach for granting access to restricted resources. Indeed as research has shown (Verizon, 2009), over half of all security breaches on computer systems are due to various forms of weak passwords. Nevertheless, alternative approaches and protocols for implementing access control have recently appeared - e.g. WebID (W3C, 2011e), OpenID (OpenID Foundation, 2011) and OAuth (OAuth, 2011) - and make use of Semantic Web technologies in order to deploy RBAC. The importance of using Semantic Web technologies for implementing access control is clearly central in today's web applications as Semantic Web technologies provide a machine-readable representation of data and thus facilitate data sharing on the web. In addition, Semantic Web technologies may assist in designing an access control system which also takes into consideration the location of the user thus combining RBAC with LBAC.

However, the current problem in adopting Semantic Web technologies for the development of web applications is the lack of widespread implementations in business environments as well as a poor understanding by non-technical users of the impact and benefits that this approach may offer.

*Research question*

The question therefore remains to what extent Semantic Web technologies can be used to implement role and location based access

control systems in a real business scenario and how they compare to a traditional approach based on user name and password. These are the very questions the current dissertation seeks to answer.

*Dissertation overview*

The remainder of this paper is organised as follows. In chapter 1 the context and the motivation of this study are outlined together with aim, objectives and deliverables of the project. Chapter 2 illustrates business requirements captured from a real client, discusses background research on RBAC and LBAC models and identifies relevant Semantic Web technologies in order to model the access control system. Chapter 3 models both RBAC and LBAC systems and describes how Semantic Web technologies are employed in the models. Chapter 4 develops two different access control systems and provides both a comparison and a critical evaluation. Chapter 5 addresses the issue of LBAC within a RBAC system. Finally, chapter 6 draws conclusions and sketches suggestions for future work.

## 1.1 CONTEXT, RATIONALE AND MOTIVATION

*Context and rationale*

The idea for this dissertation project started when I was approached by my former line manager at Nielsen[1] for recommendations on redesigning one of their databases. After discussions with her and my supervisor, we agreed to concentrate on the issue of access control since it represents the most interesting research aspect of the project. In particular, the access control system to be implemented should be able to answer the following questions:

- Which resources can an employee access?
- Which kind of access is granted on a resource?
- How to give access to resources based on the location of an employee?

The project includes a practical application in the form of a piece of web-based software. In addition, the presence of a client allows to develop the software with real business requirements and constraints.

*Motivation*

My personal choice of this area is a result of the fact that I have attended modules on Semantic Web and Web Interfaces & Media in the frame of my MSc. Indeed, this project allows me to apply concepts and programming techniques acquired during the above modules and to put them into practice in a practical project. Furthermore, the links with my former employer Nielsen provides me with an understanding of the client's business requirements which are vital for drawing specifications.

Finally, the project originality consists of applying Semantic Web technologies for developing RBAC and LBAC systems. Although this

1 Nielsen (2011) is a global marketing, information and measurement company with a seat in Oxford where I have worked prior to my MSc.

approach was used in the past, there is a lack of practical implementations in a business environment as the literature review will show. Moreover, new Semantic Web technologies have been developed in recent years and their use in RBAC and LBAC systems can provide a new approach to the problem.

## 1.2 AIM, OBJECTIVES AND DELIVERABLES

*Aim*

The aim of the project is to design and implement a role and location based access control system using Semantic Web technologies. For the purpose of this dissertation, the project only focuses on access control and does not implement other business functionalities.

*Objectives*

The objectives of the project are as follows:

- Capture business requirements and specifications at Nielsen
- Identify and evaluate existing RBAC and LBAC models
- Identify Semantic Web techniques suitable for implementing access control systems
- Design, implement, deploy and test the access control system
- Evaluate the developed system, the underlying models and the Semantic Web approach.

*Deliverables*

The following deliverables are included in the project:

A. A report detailing the following:
   - A critical evaluation of RBAC and LBAC models and their applications.
   - An analysis of Semantic Web technologies for implementing access control systems.
   - Specifications, design, implementation and risk analysis plan with a discussion of the difficulties experienced.
   - An overall evaluation of the developed system.

B. A piece of software implementing RBAC and LBAC using Semantic Web technologies.

# 2 BACKGROUND RESEARCH

This chapter illustrates business requirements for the access control system and discusses background research on RBAC, LBAC and the Semantic Web.

## 2.1 BUSINESS REQUIREMENTS

Business requirements were captured during three meetings with the Nielsen manager followed by discussions with the dissertation supervisor so that the project could be framed within an academic research.

Since the focus of the project is access control, the system is kept simple and business functionalities within the system have not been developed but are only considered when they influence the access control system.

The main techniques used for the elicitation of business requirements were interviews, questionnaires and scenarios. A great number of organisational requirements and business constraints were already known as I was familiar with the business environment, having worked at Nielsen in the past. As a result, the number of meetings could be kept to a minimum. Since the scope of the project had been limited to the development of an access control system, other techniques for capturing business requirements such as use cases or viewpoints were not employed during the business requirement phase.

The most interesting aspects emerging from interviews and scenarios were the enforcement of internal Nielsen policies and the development of additional functionalities for accessing resources on the enterprise intranet. For instance, Nielsen gives the possibility to helpdeskers to work outside the Nielsen offices only if the tasks are web-based related and if the employee has been working for the company for a certain period of time in order to build the trust necessary for a more relaxed working policy. Consequently, it is compulsory for junior helpdeskers to enter daily working hours from the Nielsen premises. Another example is the need for managers to access budgets and reports on the system while being out of office. However, since Nielsen also co-operates with competitors in few shared projects, there is a concern that sensitive information could be leaked or sneaked out at competitors' offices. Therefore, the access control system should be location aware and able to check the access position of the authorized user before granting permission on key resources.

### 2.1.1 *Roles, resources and permissions*

In order to model the main business constraints, the following job roles are identified:

- manager (line manager and helpdesk administrator)
- helpdeskers (junior and senior)
- accountant
- system administrator

The main functionalities where access is to be granted are identified as follows:

- entering working hours
- accessing business reports
- accessing payrolls and budgets
- remote access

Finally, access can be granted in order to perform the following operations:

- read (r)
- write(w) - i.e. adding new data
- edit (e) - i.e. edit existing data
- delete (d)

Finally, in order to better define the specifications, competency questions are identified so that the access control system should provide information regarding the following issues:

- Which resources can an employee access?
- Which kind of access is granted on a resource?
- How to give access to resources based on the location of an employee?

The above roles, resources and permissions are subsequently used in chapter 3 for modelling the access control system.

## 2.2 LITERATURE REVIEW ON RBAC

Role-based access control (RBAC) was first introduced in a seminal paper by Ferraiolo & Kuhn (1992) in order to provide a more suitable model for managing levels of access control in non-military systems. Its underlying model consists of mapping a set of system's permissions to roles. Consequently, individual users can be granted permissions to perform actions in a system according to their role in the organisation.

RBAC was standardized in 2004 as ANSI RBAC (ANSI, 2004) and has become the predominant access control system employed in commercial and governmental environments. However, the ANSI RBAC standard has been criticized by Li et al. (2007) due to a number of logical flaws including its openness to different interpretations of role hierarchy between software vendors. As a result, Li et al. (2007) have introduced a more formal model which is claimed to solve flaws of ANSI RBAC. More recently, Power et al. (2009) have refactored Li et al.'s model in order to provide a simpler and more modular RBAC system by further normalizing roles and permissions. However, the assessment of this RBAC model in practical implementations is somewhat limited as it has mainly been applied only to large medical systems (e.g. Simpson et al. (2005), Geddes et al. (2005), cited in Power et al. (2009)).

Another recent approach for developing RBAC systems is the use of Semantic Web technologies for defining roles and permissions. An attempt was performed by Dale (2002) who developed an access control engine for RDF and provided case scenarios for its implementation. Since Dale's work, a number of new Semantic Web technologies have been introduced. In particular, the Web Ontology Language OWL (W3C, 2011) has been recently applied in RBAC systems (*inter alia* Cirio et al. (2007), Finin et al. (2008)). However, the approaches vary considerably according to the Semantic Web technologies used. For instance, Cirio et al. (2007) employ a mix of Description Logic reasoner and SPARQL queries for inferring roles within the ontology and extracting information from the RBAC system. On the other hand, Finin et al. (2008) provide another approach which makes use of rules for modelling roles in RBAC. However, both works only provide proof-of-concept implementations of the RBAC systems and do not assess the RBAC system in a concrete and practical implementation.

*Research focus*

From the reviewed literature on RBAC, it is evident that further research is required. The question clearly remains how best to implement and deploy an RBAC system using Semantic Web technologies in a practical project for a real client. This is one of the key issues that the current dissertation seeks to investigate.

## 2.3 LITERATURE REVIEW ON LBAC

Location-based access control (LBAC) is a form of access control which takes into account the physical location of a user when granting access to resources through a mobile device (Ardagna et al., 2006). This implies the need to perform location verification on the user's physical position. The main issue on location verification is that the user can change position within seconds so that the verification method should regularly check her location. The most used technologies employed for identifying users' locations are the Global Positioning System (GPS) (Getting (1993) cited in Ardagna et al. (2006)) and signal triangulation between the mobile device and the wireless stations. Information which can be retrieved includes among others the location of the user expressed in terms of longitude, latitude and altitude as well as the detection accuracy. However, location verification is prone to error since the position retrieved by current technologies is always approximate. This is mainly due to technological limitations, environmental factors and to the fact that the user may rapidly change her position when in motion.

In order to address the issues above Ardagna et al. (2006) have developed an LBAC model taking into account the user's location as well as the confidence in the detection accuracy. This allows the development of an access control granting permission on resources only if confidence is within set thresholds. The advantages of using this model is that it integrates both location and identity-based access control making it a suitable candidate model for this project. Furthermore, the model addresses issues related to location approximation outlined above with reference to access control policies. However, the study of Ardagna et al. (2006) is limited to a theoretical model and neglects to provide a practical implementation on a web-based application.

*Research focus*

The question therefore remains how to implement such a location aware access control system, how to combine it on top of an existing RBAC system and to which extent Semantic Web technologies can be employed in implementing such a system. This is one of the issues that the current dissertation seeks to investigate.

## 2.4 LITERATURE REVIEW ON SEMANTIC WEB

The Semantic Web is often referred to as web of data where data from different sources can be related, shared and reused on the web. The term was first coined by Berners-Lee et al. (2001) and since then a great number of technologies have been introduced by the World Wide Web Consortium (W3C) and other organisations and companies. The main purpose of the Semantic Web vision is to add semantic information to data so that machines can automatically make sense of information understandable to humans.

The Semantic Web also refers to a set of technologies mainly developed within the W3C (W3C, 2011b). Semantic Web technologies are based on Resource Description Framework (RDF) allowing to represent relationships between data as triples - i.e. statements comprising a subject, an object and a predicate describing an association of some type between subject and object. Data can be stored in RDF triple stores acting as databases for storing RDF triples. Moreover, data expressed in RDF is extremely flexible as it can be serialised in XML and various other formats.

Other key technologies for representing knowledge are the Web Ontology Language (OWL) and the Friend of a Friend (FOAF) vocabulary. OWL (W3C, 2011) extends RDF as it allows reasoning and inference so that new relationships between data can be discovered whereas FOAF (FOAF Project, 2011) provides a standard vocabulary for describing persons and their relations.

The main advantage of using Semantic Web technologies is their ability to link information from multiple sources, to provide a semantic representation of data and to allow reasoning so that information from different sources can be merged and linked. As a result, data can be easily reused across the web and new relations between data not represented in any previous source can arise.

With regard to access control, Semantic Web technologies offer the advantage to align and share various access control policies between different departments of the same company or organisation thus providing a common shared model for accessing restricted resources (Finin et al., 2008).

Although Semantic Web technologies are a powerful tool for expressing knowledge, they may also expose confidential data on the web and infer new unexpected concepts. In particular, the use of Semantic Web technologies in the Social Web may constitute a risk in terms of privacy. In fact, sharing data on the web is an asset only if users are aware of it and keep control of their own data. In order to address these issues, recent advances on the Semantic Web have seen the development of decentralized authentication protocols allowing users to share data on the web while remaining in control of what they share and with whom. Such protocols include WebID (W3C, 2011e), OpenID (OpenID Foundation, 2011) and OAuth (OAuth, 2011) and are discussed in more detail in chapter 3. In particular, the WebID protocol makes use of FOAF for authenticating users and therefore represents an interesting use of the Semantic Web for enforcing access control policies. However, its deployment in business applications is still underused and the question remains how it compares to a more traditional approach for authenticating users on restricted areas of a website. This is one of the issues that the current dissertation seeks to investigate.

*Research focus*

## SUMMARY

This chapter has illustrated the business requirements for the project and outlined literature reviews on RBAC, LBAC and Semantic Web. The following chapter models the access control system.

# 3 MODELLING

This chapter describes the modelling of both the RBAC and LBAC systems and illustrates how Semantic Web technologies are employed in the models. The chapter concludes with a risk analysis and the elicitation of criteria in order to evaluate the success of the project.

## 3.1 RBAC MODELLING

### 3.1.1 *Role Hierarchy*

According to the specifications outlined in chapter 2, the role hierarchy with the corresponding permissions on resources is illustrated in figure 3.1. Permissions are declared using properties `Permitted` or `Prohibited` and access rights on resources are defined as `r(ead)`, `w(rite)`, `e(dit)` and `d(elete)`.

Roles within the company are organised through an inheritance mechanism. On the one hand, this allows the sharing of common permissions between two similar roles (e.g. junior and senior helpdeskers). On the other hand, more refined permissions for specific roles can be defined through subroles. For instance, all managers share read access right on reports but only line managers have write, edit and delete access on this resource. For the sake of clarity, unauthorised access rights are explicitly declared with property `Prohibited`; however, this is partially redundant as the lack of property `Permitted` on a resource would suffice to deny any access type on that resource. Finally, access on resource remote does not bear any access type as this resource simply allows remote access to a computer.

### 3.1.2 *Ontology design*

RBAC is modeled into an OWL ontology following the second approach outlined in Finin et al. (2008). In particular, this approach models roles as values and is preferred to that modelling roles as classes as it is better suited for this project and its representation is simpler.

Protégé-OWL 4.1 (Stanford Center for Biomedical Informatics Research, 2011b) is employed to build and edit the OWL ontology. The main classes of the RBAC system are identified as follows:

- `Action` representing actions that a user can perform. This class contains two subclasses `PermittedAction` and `ProhibitedAction` which are disjoint.

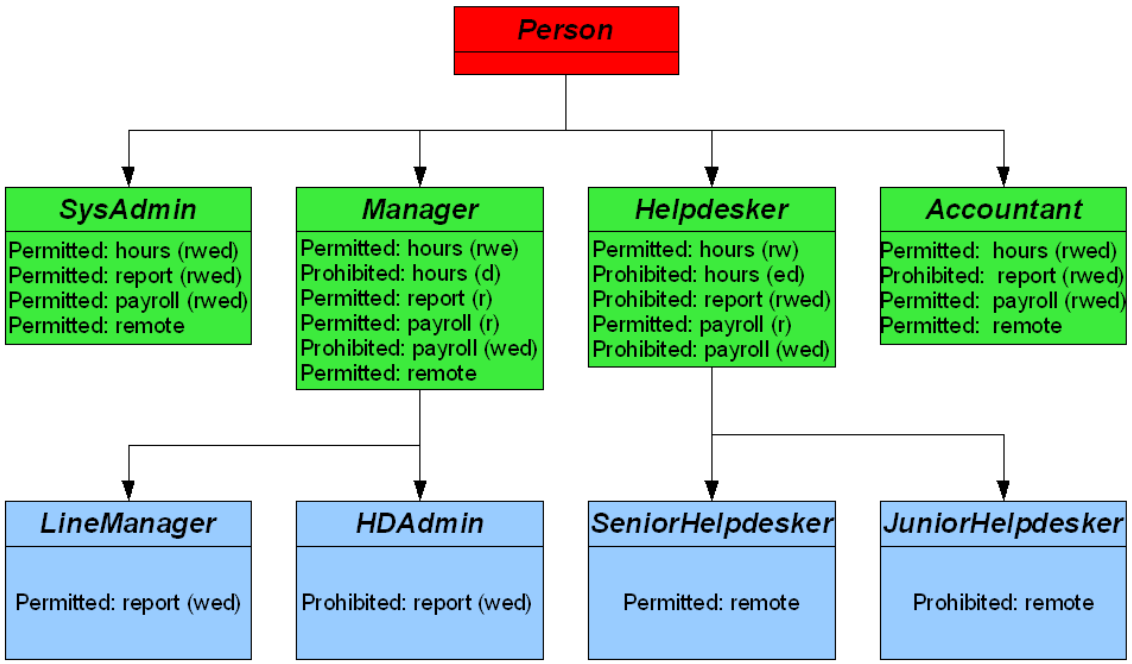


Figure 3.1: Role hierarchy.

- `Role` representing the roles that a user can have.
- `Person` representing Nielsen employees.
- `Resource` representing resources which can or cannot be accessed.

Classes with instance data are represented in Figure 3.2[1].

Permissions and access types are bundled together by means of blank nodes. A blank node is an RDF node without any Uniform Resource Identifier (URI) or literal value so that it can act as a container for other URIs. In fact, its purpose is to link a resource to one or more access types for a given user. Figure 3.3 illustrates a graphical representation of a blank node[2] whereas Listing 3.1 shows the corresponding RDF code.

Table 3.1 and 3.2 describe a basic set of object and data properties where `dns` accounts for the default namespace and `foaf` for the imported FOAF vocabulary (FOAF Project, 2011). Other properties can be added in order to provide more information about the classes. The complete OWL code is listed in Appendix A

Although a Web Access Control ontology was developed in the frame of the W3C (2011c), this has not been used since it is too generic for the purpose of this project. In particular, the ontology does not

1 Although all users are linked to roles, Protégé's visualisation tool somehow misses a relation linking user `sandro` to role `juniorhelpdesker`.

2 For the sake of brevity the default name space in the graphical representation is abbreviated to `http://dns#`.

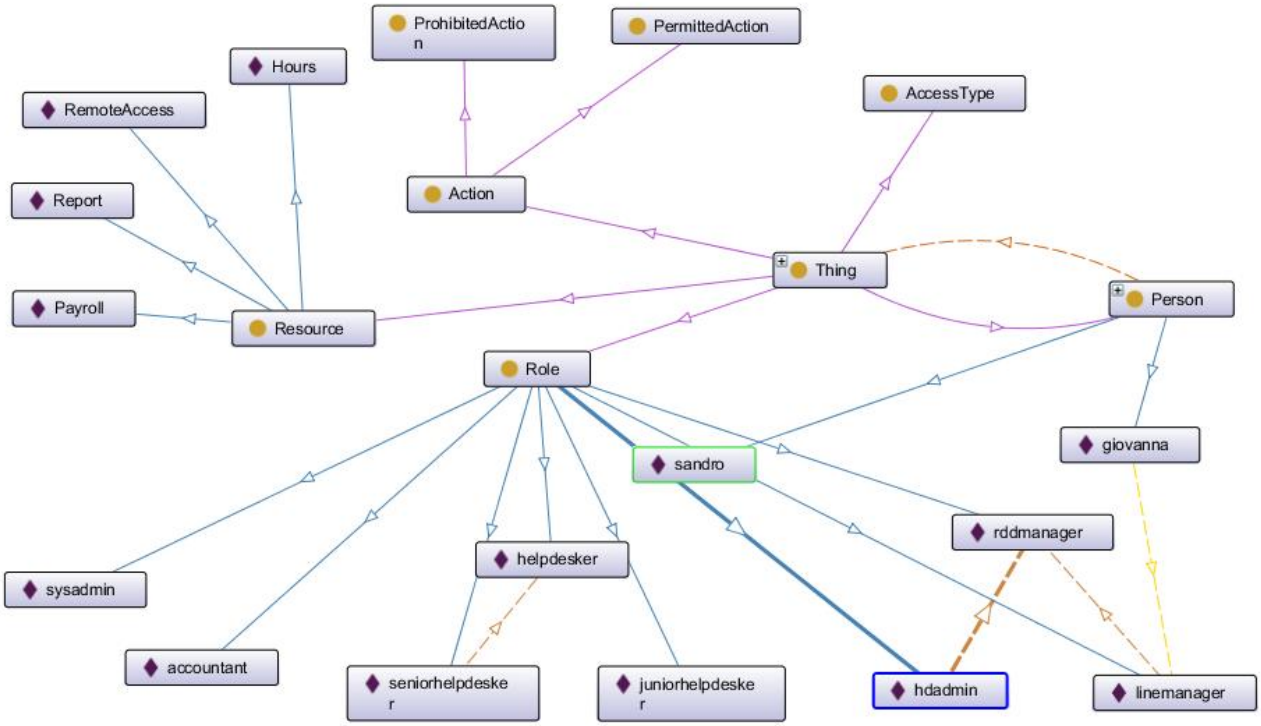


Figure 3.2: Graph representing classes and instance data

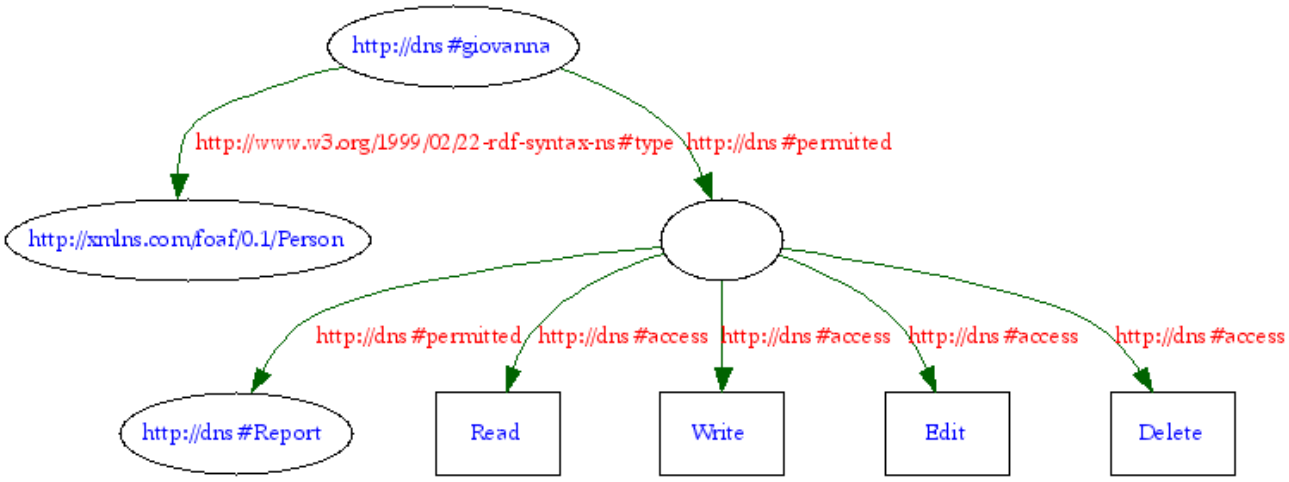


Figure 3.3: Graphical representation of a blank node

```
<foaf:Person rdf:about="http://www.sandrocirulli.net/
    dissertation/ontology/ontology.owl#giovanna">
  <permitted rdf:nodeID="ac1" />
</foaf:Person>

<rdf:Description rdf:nodeID="ac1">
  <permitted rdf:resource="http://www.sandrocirulli.net/
      dissertation/ontology/ontology.owl#Report"/>
  <access rdf:datatype="&rdfs;Literal">Read</access>
  <access rdf:datatype="&rdfs;Literal">Write</access>
  <access rdf:datatype="&rdfs;Literal">Edit</access>
  <access rdf:datatype="&rdfs;Literal">Delete</access>
</rdf:Description>
```

Listing 3.1: RDF blank node

| PROPERTY | DESCRIPTION | DOMAIN | RANGE | CHARACT. |
|---|---|---|---|---|
| dns:grantedOn | Permission granted on | Action | Resource | Functional |
| dns:grantedTo | Permission granted to | Action | Person | Functional |
| dns:hasRole | Possible role | Role | Role | |
| dns:hasActiveRole | Activated role | Person | Role | |
| dns:isSubRole | Subrole for defining role hierarchy | Role | Role | subproperty of dns:hasRole |
| dns:permitted | Permission to perform task | Role | Action | |
| dns:prohibited | Prohibition to perform task | Role | Action | |
| foaf:knows | Friendship | Person | Person | Symmetric |
| foaf:mbox | Personal mailbox | Person | | Inverse functional |
| foaf:mbox_sha1sum | sha1sum of mailbox | Person | | Inverse functional |
| foaf:phone | Phone number | Person | | |

Table 3.1: Object properties

| PROPERTY | DESCRIPTION | DOMAIN | RANGE |
|---|---|---|---|
| foaf:family_name | Surname | Person | Literal |
| foaf:givenname | Firstname | Person | Literal |
| foaf:name | Full name | Person | Literal |
| foaf:gender | Gender | Person | Literal |

Table 3.2: Data properties

take into account roles and does not bind resources with the type of access; indeed, these issues are still under discussion within the W3C (W3C, 2011d). Finally, it is interesting to notice that the development of semantic Access Control List (ACL) is an area of current research (*inter alia* Hollenbach et al. (2009; 8) and MyProfile (2011b)).

### 3.1.3 *Authentication and authorization*

The concepts of authentication and authorization are of paramount importance in any access control system and must not be confused.

*Authentication*

Authentication refers to the action of verifying the identity of the user and checks user's credentials to access the system - e.g. user name and password, card and PIN, fingerprint, security token, etc.

*Authorization*

Authorization is defined as the process of verifying which sets of permissions and access types are granted to an authenticated user. As a result, authentication always precedes authorization.

For this project, the process is somehow complicated by the need to check both the role and the location of the user. Therefore, the authentication and authorization processes are performed in the following order:

A. *Authentication*

B. *Authorization*

   a) Verifying *role-based* conditions

   b) Verifying *location-based* conditions

Issues regarding location-based conditions are discussed in paragraph 3.2.

## 3.2 LBAC MODELLING

LBAC is represented following the model provided by Ardagna et al. (2006). This theoretical model is of particular interest since it takes into account both the user's location and the confidence in the detection accuracy. In fact, the location is always expressed as a range rather than an exact value in order to account for approximation in the location detection. For instance, a query asking for the user's location returns a triple made up of values for location range, accuracy of the location provided and time frame within which values can be considered valid. The query is modeled as function of the form

$$predicate(parameters) -> [range, accuracy, timeout]$$

For the sake of verifying permissions on a resource, the query is adapted for returning a boolean value and a confidence level so that access policies can be applied. Therefore the query is modeled as function of the form

$$predicate(parameters, value) -> [bool\_value, confidence, timeout]$$

thus returning a boolean value, a confidence percentage and a timeout period.

Modelling predicates assumes the presence of elements such as `User` identified by a SIM card and `Area` mapping regions which can range from a room to a city. These elements are employed in predicates for evaluating a specific statement. Among the predicate expressions analysed by Ardagna et al. (2006), the following is of particular interest since it verifies whether a user is within an area as well as the confidence level assigned to the detection:

$$inarea(user, area) -> [bool\_value, confidence, timeout]$$

For instance, let `Giovanna` be an element of `User` with a managerial role and `CompetitorOffice` be an element of `Area`. A typical access control scenario based on location for this project is:

$$inarea(Giovanna, CompetitorOffice)$$

$$-> [True, 0.96, 2011\text{-}09\text{-}02\_15{:}00pm]$$

stating that it is true that manager Giovanna is at a competitor's office with an accuracy of 96% and that this statement is valid until 3pm on September, 2nd. Should the accuracy be below a given confidence threshold, the location is detected once again. In order to avoid deadlocks, a maximum number of tries for location detection is set in advance.

Finally, rules based on predicate expressions may reflect the following access control policy at Nielsen:

1. A manager cannot access reports from a competitor's office.
2. A senior helpdesker can access any resources linked to her role from home or from the Nielsen office but not from a competitor's office.
3. A junior helpdesker can access resource hours only from the Nielsen office.

### 3.2.1 *LBAC scenarios*

As outlined in paragraph 3.1.3, a mix of role-based and location-based conditions is taken into account in the access control system. In particular, location-based conditions reflect the inaccuracy of location verification so that access is granted only if the location is above a certain confidence level. The implementation may be extended in order to evaluate other predicates taking into account factors such as altitude, velocity of motion and presence of other users within an area.

| TYPE | PREDICATE | DESCRIPTION |
|---|---|---|
| Position | `inarea`(*user, area*) | Evaluate whether *user* is located within *area* |
| | `disjoint`(*user, area*) | Evaluate whether *user* is outside *area* |
| Movement | `velocity`(*user, min_vel, max_vel*) | Evaluate whether *user*'s speed falls within range [*min_vel, max_vel*] |
| Interaction | `density`(*area, min_num, max_num*) | Evaluate whether the number of users currently in *area* falls within interval [*min_num, max_num*] |

Table 3.3: Location-based predicates [adapted from Ardagna et al. (2006)].

Table 3.3 illustrates the main location-based predicates for this project whereas table 3.4 outlines three basic access control rules regulating access to the system. Role-based conditions are verified first and the authentication process is either performed via user name and password or through digital certificates as explained in paragraph 3.4. After that, location-based predicates are employed to verify location conditions. The implementation of the above scenarios are discussed in more detail in chapter 4 for role-based conditions and chapter 5 for location-based conditions.

One of the main issues of mixing role and location based conditions is that role conditions are generally static - i.e. they do not change in the long run - whereas location conditions are dynamic - i.e. they might change at any time. Although location conditions restrict the permissions given to a role, they should still allow access to resources which are not influenced by location. For instance, according to the access control rule in the first scenario, a manager cannot access reports anymore due to the fact that she is in a competitor's office; however, she should still be able to access other less sensitive resources such as remote, hours and payroll. In order to overcome this issue, two complementary approaches can be employed:

A. Static approach - each possible role is wired within the role hierarchy at compile time.

B. Dynamic approach - access to resources is filtered at runtime so that temporary restrictions can be applied.

The static approach adopts a close world assumption so that all possible roles must be described in the role hierarchy at compile time, thus allowing the thorough description of each role and its

| # | ROLE CONDITIONS | LOCATION CONDITIONS | ACTION | DESCRIPTION |
|---|---|---|---|---|
| 1 | user.Role=Manager ∧<br>Valid(user.Username, user.Password) ∨<br>Valid(user.ID, user.Certificate) | disjoint(user, CompetitorOffice) | report(rwed) | Managers are authorized to access reports if they are not in a competitor's office. Action applies to both subroles LineManager and HDAdmin |
| 2 | user.Role=Helpdesker ∧<br>Valid(user.Username, user.Password) ∨<br>Valid(user.ID, user.Certificate) | inarea(user, NielsenOffice) ∧<br>velocity(user, 0, 3) | hours(rw) | Helpdeskers are authorized to enter working hours if they are in the Nielsen office and move at walking speed at most. |
| 3 | user.Role=Manager ∧<br>Valid(user.Username, user.Password) ∨<br>Valid(user.ID, user.Certificate) | local_density(user, Close By, 1, 1) ∧<br>disjoint(user, CompetitorOffice) | report(rwed) | Line managers are authorized to fully access reports if there is nobody close by and they are not in a competitor's office. |

Table 3.4: Access control rules regulating access [adapted from Ardagna et al. (2006)].

corresponding permissions. However, this approach lacks scalability since new location-based conditions may generate further roles and complex sets of permissions which were not foreseen during the design phase. As a result, the role hierarchy would need to be adapted or redesigned each time a new location-based rule is implemented.

Conversely, the dynamic approach adopts an open world assumption so that new roles can be taken into account at runtime through filtering of permissions. This approach allows dynamic access to resources so that new location-based rules can be implemented without modifying the underlying role hierarchy. Another advantage of this approach is that it allows the temporary restriction of the access to resources for a single user without dropping her from the list of users. This may well be the case when permissions of a user must be temporarily deactivated due to abuse, change of department or the like. As a result, the dynamic approach tends to be more flexible than the static approach and is therefore used in the implementation shown in chapter 5.

## 3.3 AUTHENTICATION METHODS

The most popular authentication method is based on a combination of user name and password. Notwithstanding its popularity in current web applications and computer system, the use of this traditional authentication method is often criticised for the following reasons:

- *Security* - allowing users to choose passwords may result in the use of weak passwords or a unique password for all services, thus risking the security of the whole system.
- *User-friendliness* - Users may have to remember multiple passwords. As a result, they tend to use the same password for different services or to write them down in an unsecured manner.

Recent development in authentication protocols have seen the development of alternative authentication methods. The most important are identified and described as follows:

- OAuth (2011) is an open protocol for authentication that uses tokens for granting access to resources on different websites.
- OpenID (OpenID Foundation, 2011) is an open protocol for identifying a user in a decentralized way by means of a unique URI.
- WebID (W3C, 2011e) is an open protocol for identifying a user in a decentralized way using the combination of a FOAF file and a digital certificate. This protocol is also known as FOAF+Security Sockets Layer (SSL).

OAuth was first introduced in 2006 and is now widely supported by APIs of popular web services including Google, Facebook and Twitter. However, the current OAuth 2.0 drops the backward compatibility with the previous version and has been recently criticised on security grounds for its use of tokens (Dubray (2011), Adida, B. (2011)).

OpenID dates back to 2005 and is also an established and widely supported protocol. However, it only provides a decentralized identification on the web and cannot make use of linked data (Hollenbach et al., 2009; 3).

WebID is a more recent protocol since it has been introduced in the past two years (Story et al., 2009). Nevertheless, it is extremely flexible and reusable since it can be easily mapped to an OpenID account (W3C, 2011a). Furthermore, it offers various advantages over OpenID as it allows the association of linked data through a FOAF file as well as a more distributed approach (W3C, 2011f) due to its RESTful architecture[3]. Finally, it is the protocol that more heavily relies on Semantic Web technologies - most notably FOAF - and is now developed within W3C thus making it a suitable authentication protocol for this dissertation project. However, it should be noticed that WebID is still a young protocol and at the time of writing is less used by major web services when compared to OAuth and OpenID.

### 3.3.1 *How WebID works*

The FOAF+SSL / WebID protocol authenticates a digital identity (WebID) by allowing a web browser to prove the possession of a private key, whose public key is bound to the WebID (W3C, 2011e).
In other words, the authentication process is performed by checking asymmetric cryptographic keys on a digital certificate and on a public URI so that one-click sign-on to web services is allowed.

The core components of the WebID protocol are:

- *FOAF* - Friend of a Friend vocabulary for describing a person and her relations.
- *SSL* - Secure Sockets Layer for providing secure and encrypted data transmission over the web
- *WebID* to uniquely identify a person using a URI.
- *Digital certificates* using public-key cryptography.

The use of FOAF+SSL / WebID requires the creation of an X509 certificate, a FOAF file and a WebID. A digitally signed X509 certificate is an encrypted certificate employing public key cryptography to match a digital identity with a public key. Such a certificate can

3 A web service is said to be RESTful when it implements Representational State Transfer (REST)'s design principles (Fielding, 2000) .

be easily generated using freely available command line tools such as OpenSSL (The OpenSSL Project, 2011). Similarly, a FOAF profile containing information about a user can be created using online tools such as MyProfile (2011a). In addition to creating a FOAF file, MyProfile can also automatically link the profile to a WebID in order to uniquely identify a user on the web.

The mapping of the X509 certificate with the FOAF profile is performed during the creation of the certificate. In fact, the FOAF URI that uniquely identifies a user must be included in the certificate. Detailed instructions on how to create an X509 certificate and link it to both a FOAF file and a WebID are provided by Story (2011a).

Listing 3.2 shows a decoded X509 certificate using three relevant key fields for this implementation such as the Issuer field describing the details of the issuing authority, the Subject field describing the common name (CN) of the certificate owner, and the Subject Alternative Name providing the WebID. Encrypted values have been omitted for the sake of brevity.

The UML diagram in table 3.4 illustrates how the FOAF+SSL / WebID protocol works for authentication and authorization. Sandro's browser requests a page available on the Nielsen's public area website which is then successfully returned by the server (1). After that, Sandro's browser requests a page stored in the Nielsen's protected area. The connection is performed using SSL which performs a handshake procedure and verifies that Sandro's browser is in possession of a certificate and its corresponding private key. The certificate containing the public key and the URI where the FOAF file is stored is sent to server (2). The server retrieves Sandro's FOAF file (3) and verifies that public keys on the provided certificate and the FOAF file match (4). If so, the user is authenticated and the requested page is retrieved through a SPARQL query from the ontology (5). Finally, the page is returned to the user's browser (6).

## 3.4 USE OF SEMANTIC WEB TECHNOLOGIES

Table 3.5 describes the development infrastructure and the Semantic Web technologies employed to develop this project as well as a justification of the technologies used. In case of equivalent technologies, the choice was mainly personal and based on previous knowledge of the technology acquired during module P00775 Semantic Web. As the table shows, this is a pure Java solution based on the currently most popular tools for developing Semantic Web applications.

## 3.5 RISK MANAGEMENT

Risk management plans have been developed at the early stages of the project in the frame of the P00011 module (March 2011), the project

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f5:8a:b2:d1:76:06:13:8f
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=FR, ST=Essonne, O=webid.fcns.eu, CN=webid.fcns.eu/emailAddress
=webid@fcns.eu
        Validity
            Not Before: Jul 10 14:28:27 2011 GMT
            Not After : Jul  9 14:28:27 2012 GMT
        Subject: CN=Sandro Cirulli
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:c2:2d:26:2a:ac:0f:87:4e:e0:ef:e9:51:ac:e0:
              [..]
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                B6:36:80:6A:C1:60:C1:32:4B:70:A1:0D:DE:F8:68:C4:36:E9:AD:74
            X509v3 Authority Key Identifier:
                keyid:2B:DF:EF:BF:79:13:73:CB:E4:D4:35:A5:0B:EC:18:2C:63:E4:D2:F0
            X509v3 Subject Alternative Name:
                URI:http://webid.fcns.eu/people/SandroCirulli/card#me
    Signature Algorithm: sha1WithRSAEncryption
        a9:23:0d:7e:5e:a4:2c:fb:ee:b9:74:b8:86:b1:b7:ee:dc:46:
    [..]
```

Listing 3.2: X509 certificate

| COMPONENT | TECHNOLOGY CHOSEN | ALTERNATIVE TECHNOLOGIES | JUSTIFICATION |
|---|---|---|---|
| Programming language | Java | | Most Semantic Web technologies are freely available for Java. |
| Scripting languages | JSP, Servlets & JavaScript | CGI scripting (Perl, PHP, etc.) | CGI scripts have drawbacks in terms of performance and interaction with the server (Hunter, 2001; 2–3). |
| Web server | Tomcat | Glassfish, JBoss | Tomcat is the official implementation for JSP and Java Servlets (Apache Software Foundation, 2011b). However, any other Java-based server supporting servlets would have been equivalent. |
| Ontology editor | Protégé - OWL 4.1 | Protégé 3.4 | Protégé 4.1 fully supports OWL 2 and is more suited than version 3.4 for developing ontologies purely written with OWL (Stanford Center for Biomedical Informatics Research, 2011a). |
| Semantic Web Framework & Triplestore | Jena | Sesame | Both frameworks support Java and would be equivalent. |
| Query language | SPARQL | | |

Table 3.5: Development infrastructure

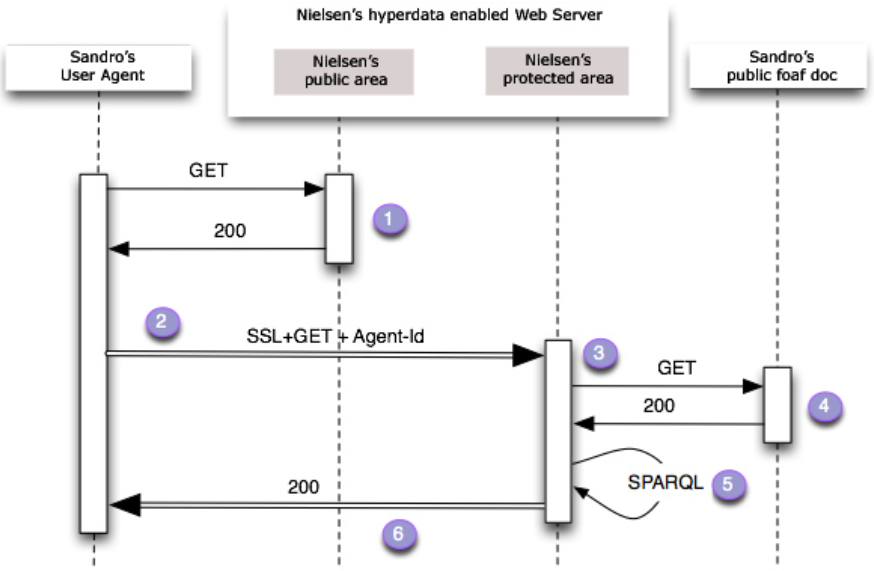


Figure 3.4: FOAF+SSL / WebID protocol [adapted from Story et al. (2009)].

proposal (March 2011) and the interim report (April 2011) in order to tackle prospective risks. As recommended by Dawson (2005), risk management encompasses the identification of risks, assessment of risks impact, strategies for alleviating critical risks and for controlling risks. As a result, the following documents have been produced:

- Risk management plan
- Time frame for activities and dependencies
- Gantt chart (2 versions)
- PERT chart
- Risk assessment form
- Risk severity matrix
- Risk response matrix

It is worth to noticing that the Gantt chart describing the project schedule has been reviewed twice in order to take into account changes and progress of the project. Risk assessment form, severity and response matrices are included in Appendix B for reference.

## 3.6 EVALUATION CRITERIA

Chapter 4 and 5 describe how the access control system is developed and implemented. In order to evaluate and measure the success of the project, the following evaluation criteria are made explicit:

- Two access control systems based respectively on user name and password and WebID should be developed and subsequently compared and contrasted.
- The access control systems should be able to answer competency questions taking into account roles, permissions and access types as outlined in chapter 1
- The preferred access control systems should include LBAC and concurring dynamic factors based on the position.

As a result, the success of the project is measured by the ability of the system to address the above issues and by the granularity that the access control system can provide.

### SUMMARY

This chapter has described the modelling of the dissertation project and discussed issues such as RBAC and LBAC modelling, description of authentication methods, use of Semantic Web technologies, risk management and evaluation criteria. The following chapter outlines the step undertaken to develop a RBAC system.

# 4 ROLE BASED ACCESS CONTROL

This chapter describes the development and implementation of two access control systems respectively based on form authentication - i.e. user name and password - and the WebID protocol. The chapter concludes with a comparison and an evaluation of both systems.

## 4.1 SOFTWARE DEVELOPMENT

The code of both access control systems is available as NetBeans projects on the companion CD and has been developed using NetBeans IDE on Linux Ubuntu 10.04 and Windows XP on the pooled computers at Brookes University. The software developed reuses and modifies source code available under the Apache Licence Version 2.0 (Apache Software Foundation, 2011a) and employs Semantic Web technologies released under various open source licences. As a result, all the source code developed in this project is released and can be redistributed under the above Apache Licence.

As a convention, `root` is used for identifying the root of the project so that the systems can be replicated by pasting all files in a new root directory and recompiling the project. Alternatively, the Java web application can be deployed as .war file from a fresh Tomcat installation. This requires coping the .war file into the `$CATALINA_HOME/webapps` directory[1] so that starting Tomcat will automatically expand and deploy the application. The .war file is available at `root/NetbeansProjects/JSPExamples(J2EE1.4)/dist/TomcatJSPExample.war`. Both deployment system require the inclusion of Jena libraries available in the companion CD at `root/ARQ-2.8.8/lib`.

Tomcat is set up to be runnable from localhost from both ports 8080 using the HTTP protocol and 8443 using the HTTPS protocol. It should be noted that running the web applications from NetBeans may cause an error message of the kind

```
java.lang.IllegalArgumentException:
URI has an authority component
```

This is probably due to namespace issues or path resolution; however, the exception throwing does not influence the correct running of the web applications.

1 Following Tomcat jargon, `$CATALINA_HOME` identifies the root directory where Tomcat is installed.

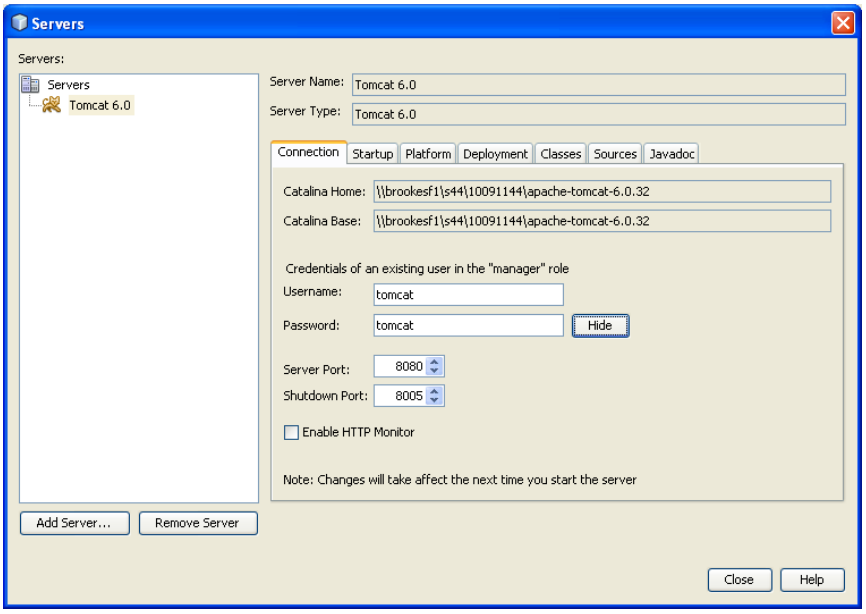


Figure 4.1: Tomcat setup via the NetBeans plugin.

## 4.2 FORM AUTHENTICATION ACCESS CONTROL

The first access control system developed is based on form authentication which makes use of user name and password as user's credentials for accessing a restricted area of a website. The following subsections describe the configuration of the Tomcat server for managing roles, users and permissions, a walk-through of the main system functionalities, and a test plan for testing the access control system.

### 4.2.1 *Tomcat configuration*

Tomcat 6.0 is set up directly from NetBeans 6.8 using the server plugin as shown in figure 4.1. For simulation purposes, a weak password is used and subsequently shown in the server configuration.

A login functionality is developed adapting the example provided in the Tomcat installation and available at `http://localhost:8080/examples/jsp/security/protected/index.jsp` from a fresh Tomcat installation. This example is modified in order to support form-based authentication. This shows an authentication form asking the user for a user name and a password as described in Brittain & Darwin (2003; 43–45). The form authentication settings are implemented in `root/NetbeansProjects/form-auth/web/WEB-INF/web.xml` as shown in the next page. This specifies a login page (`login.jsp`) for authenticated users and an error page (`error.jsp`) in case the authentication process fails.

```
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>Example Form-Based Authentication Area</realm
      -name>
  <form-login-config>
  <form-login-page>/security/protected/login.jsp</form-
      login-page>
  <form-error-page>/security/protected/error.jsp</form-
      error-page>
  </form-login-config>
</login-config>
```

Listing 4.1: FORM authentication settings in `web.xml`

The list of roles, subroles and users is coded in `root/apache-tomcat-6.0.32/conf/tomcat-users.xml` as shown in listing 4.2. Following the role hierarchy modelling, users can have multiple roles. Users' passwords are digested using the MD5 algorithm as described in Brittain & Darwin (2003; 42–43) but are reproduced in clear in the comments for simulation purposes.

The implementation is kept simple since user names and passwords are hand coded in `tomcat-users.xml` and the file is loaded into memory when Tomcat is started up. This is done by setting up the Tomcat `UserDatabaseRealm` realm[2] in `root/apache-tomcat-6.0.32/conf/server.xml`. In a real deployment, it would be recommended to dynamically access data from a database or an ontology. This can be easily extended by setting up other realm implementations such as `JDBCRealm` and `JNDIRealm` in `server.xml` (cf. Brittain & Darwin (2003; 34–37)).

Authorized access to restricted resources is managed by setting up security constraints in the web application configuration file stored in `root/NetbeansProjects/form-auth/web/WEB-INF/web.xml`. In fact, a security constraint grants access to a file or a directory only to users sharing the same role. Moreover, security constraints take into account multiple roles as well as permissions inheritance between roles. For instance, according to the role hierarchy a user with subrole `linemanager` automatically inherits permissions for role `rddmanager`. As a result, it can access all the resources for `linemanager` as well as additional resources for `rddmanager`. On the other hand, a user with subrole `hdadmin` does not have any additional access to resources so that she can only access resources related to role `rddmanager`. Listing 4.3 shows security constraints applied to roles `rddmanager` and `linemanager`.

Finally, the ontology is deployed at `http://www.sandrocirulli.net/dissertation/ontology/ontology.owl` and acts as a database for retrieving roles, user names and permissions whereas Tomcat manages these data in order to grant or deny access to resources.

2 A realm in Tomcat is a collection of roles, user names and passwords.

```
<tomcat-users>
    <role rolename="tomcat"/>
    <!--  roles -->
    <role rolename="sysadmin"/>
    <role rolename="rddmanager"/>
    <role rolename="helpdesker"/>
    <role rolename="accountant"/>
    <!--  subroles -->
    <role rolename="linemanager"/>
    <role rolename="hdadmin"/>
    <role rolename="seniorhelpdesker"/>
    <role rolename="juniorhelpdesker"/>
    <!-- users -->
    <user password="de2f15d014d40b93578d255e6221fd60" roles
        ="helpdesker,seniorhelpdesker" username="mario"/>
    <!-- psw: sandro -->
    <user password="2c40837dd4b97fd00a5a598a8a25426f" roles
        ="helpdesker,juniorhelpdesker" username="sandro"/>
    <!-- psw: sandro -->
  <user password="79d2d812bf677287382b68106237b5ee" roles="
      rddmanager,linemanager" username="giovanna"/>
    <!-- psw: giovanna -->
    <user password="b5f3729e5418905ad2b21ce186b1c01d" roles
        ="rddmanager,hdadmin" username="ed"/>
    <!-- psw: ed -->
    <user password="3e7a517843ed19a2b058bd7ce723fb49" roles
        ="accountant" username="svetlana"/>
    <!-- psw: svetlana -->
    <user password="6ae199a93c381bf6d5de27491139d3f9" roles
        ="sysadmin" username="richard"/>
    <!-- psw: richard -->
    <user password="1b359d8753858b55befa0441067aaed3" roles
        ="tomcat,manager,admin" username="tomcat"/>
</tomcat-users>
```

Listing 4.2: Form Authentication in `tomcat-users.xml`

```
<!-- added security constraint for rddmanager -->
<security-constraint>
  <web-resource-collection>
  <web-resource-name>Resources Protected</web-resource-name>
    <!-- Define the context-relative URL(s) to be protected -->
    <url-pattern>/security/protected/salary/SalaryServlet/*</url-pattern>
    <url-pattern>/security/protected/report/report-read.jsp</url-pattern>
    <url-pattern>/security/protected/remote/*</url-pattern>
    <url-pattern>/security/protected/payroll/payroll-read.jsp</url-pattern>
    <url-pattern>/security/protected/hours/hours-read.jsp</url-pattern>
    <url-pattern>/security/protected/hours/hours-write.jsp</url-pattern>
    <url-pattern>/security/protected/hours/hours-edit.jsp</url-pattern>
    <!-- If you list http methods, only those methods are protected -->
    <http-method>DELETE</http-method>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
  </web-resource-collection>
  <auth-constraint>
    <!-- Anyone with one of the listed roles may access this area -->
    <role-name>rddmanager</role-name>
  </auth-constraint>
</security-constraint>

<!-- added security constraint for linemanager (subrole of rddmanager)-->
<security-constraint>
  <web-resource-collection>
  <web-resource-name>Resources Protected</web-resource-name>
    <!-- Define the context-relative URL(s) to be protected -->
    <url-pattern>/security/protected/report/*</url-pattern>
    <url-pattern>/security/protected/payroll/payroll-read.jsp</url-pattern>
    <!-- If you list http methods, only those methods are protected -->
    <http-method>DELETE</http-method>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
  </web-resource-collection>
  <auth-constraint>
    <!-- Anyone with one of the listed roles may access this area -->
    <role-name>linemanager</role-name>
  </auth-constraint>
</security-constraint>
```

Listing 4.3: Security constraints for managerial roles in `web.xml`

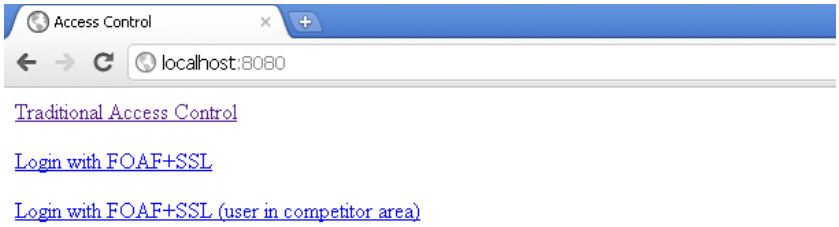


Figure 4.2: Access control menu.


Figure 4.3: Login page.

### 4.2.2 *Walk-through*

The code for the form authentication control system is included in the companion CD as NetBeans project `form-auth`. This access control system is accessed by clicking on the corresponding link as shown in figure 4.2. For the sake of simplicity, the system runs on port 8080; nevertheless, this can be easily implemented on port 8443 for secured and encrypted connections[3].

After clicking on the link, the user is prompted to insert a user name and password as shown in figure 4.3. Should the credentials match those contained in `tomcat-users.xml`, roles and permissions on resources for the given user are printed on screen (figure 4.4). In case the authentication process fails, the user lands on an error page (figure 4.5).

Data printed out in figure 4.4 is processed through `root/NetbeansProjects/form-auth/web/security/protected/index.jsp`. The name of the authenticated user is retrieved using the servlet methods `request.getUserPrincipal().getName()`. Each login is automatically assigned to a session which is then retrieved using `session.getId()`.

3 Any connection transmitting sensitive data such as a password should be performed through an HTTPS connection. However, this has not been done in this example as port 8443 is used for implementing the second access control approach in paragraph 4.3.

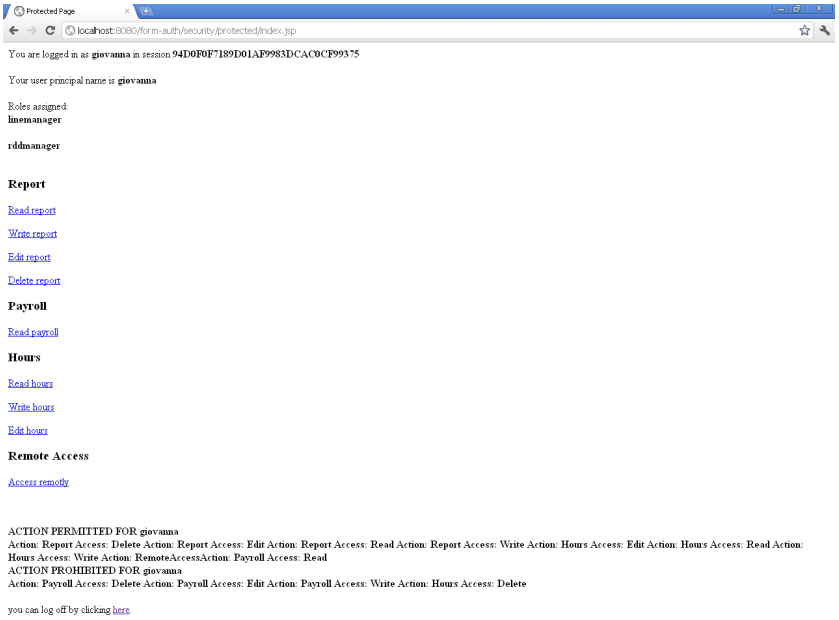


Figure 4.4: Protected page.


Figure 4.5: Error page.

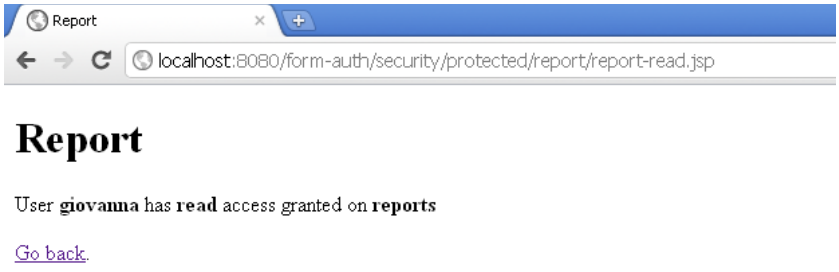


Figure 4.6: Access granted.


Figure 4.7: Access denied.

Assigned roles and subroles are retrieved from the ontology[4] using the developed Java package `accesscontrol`. This package contains a Java class allowing the creation of an ontology model and to access data from the model. The ontology is loaded by invoking the method `populateOntology()` which retrieves the ontology from the corresponding URI. Roles are retrieved by invoking method `retrieveRoles()` which runs a SPARQL query returning all roles in the ontology. Roles are then stored in an array of Strings which is subsequently traversed in order to print out only roles assigned to the given user. Once a role is retrieved, the corresponding links to resources and access types are printed out by means of the servlet method `request.isUserInRole()`. Authorized resources can then be accessed with a simple click as shown in figure 4.6. It is worth noticing that entering the URL manually in order to access an unauthorized resource generates an access denied message (figure 4.7).

Although links to resources are shown according to the corresponding role, these could also be retrieved directly from the ontology. This is achieved by calling methods `printActionPermitted()` and `printActionProhibited()` that pull out from the ontology all sets of actions permitted and prohibited with the corresponding access type.

4 This could have also been achieved using the servlet method `IsUserInRole()` that retrieves roles directly from `tomcat-users.xml`. However, since the ontology acts as a database, it was preferred to use the above procedure in order to show that roles can be retrieved and dynamically modified from a file deployed on the web.

For the sake of brevity, data in the ontology are provided for user giovanna only.

More details about the technical implementation can be found in the fully documented code in NetBeans project `form-auth`.

### 4.2.3 *Test plan*

The above access control is tested with Google Chrome 12 and the results are provided in the test plan in table 4.1.

## 4.3 ACCESS CONTROL USING FOAF+SSL - WEBID

The second access control system developed in this dissertation project is based on the FOAF+SSL protocol and makes use of digital certificates to authenticate and authorize users with a single click. The following subsections describe the configuration of the Tomcat server, a walk-through for the authentication and authorization processes, and a plan for testing the application.

### 4.3.1 *Tomcat and browser configurations*

Tomcat settings for configuring SSL and implementing FOAF+SSL imply the creation of a server certificate and the implementation of a connector. In addition, browser settings require the creation of users' certificates and subsequently their import into a browser. The procedures for both settings are described below following the configuration procedures outlined in Apache Software Foundation (2011c), Story (2011b) and Brittain & Darwin (2003; 162–172).

In order to use digital certificates with Tomcat, it is necessary to add a suitable connector for port 8443. This is done in the Tomcat configuration file `server.xml` as shown below:

```
1 <Connector
2 port="8443" protocol="HTTP/1.1" SSLEnabled="true"
3 maxThreads="150" scheme="https" secure="true"
4 keystoreFile="H:/apache-tomcat-6.0.32/conf/keystore"
      keystoreType="JKS" keystorePass="changeit"
5 SSLImplementation="org.jsslutils.extra.apachetomcat6.
      JSSLutilsImplementation"
6 acceptAnyCert="true" sslProtocol="TLS" clientAuth="false"
      />
```

Listing 4.4: Connector for FOAF+SSL

The connector allows the sending of a server certificate to a client thus identifying the server. In this implementation a self-signed certificate is used and placed in the corresponding path as specified in the value of the `keystoreFile` attribute. The certificate keystore is created

| FUNCTIONALITY | TEST INPUT | EXPECTED RESULT | ACTUAL RESULT | COMMENT |
|---|---|---|---|---|
| Login linemanager | User: giovanna<br>Password: giovanna | Accessible resources for roles rdmanager and linemanager | Correct | Permissions printed out from the ontology |
| Login hdadmin | User: ed<br>Password: ed | Accessible resources for roles rdmanager only | Correct | |
| Login juniorhelpdesker | User: sandro<br>Password: sandro | Accessible resources for roles helpdesker only | Correct | |
| Login seniorhelpdesker | User: mario<br>Password: mario | Accessible resources for roles helpdesker and seniorhelpdesker | Correct | |
| Login accountant | User: svetlana<br>Password: svetlana | Accessible resources for role accountant | Correct | |
| Login accountant | User: richard<br>Password: richard | Accessible resources for role sysadmin | Correct | |
| Authentication denied | User: test<br>Password: test | Authentication fails | Correct | |
| Authorization denied | URL from address bar for non authorized resource | Access denied | Correct | |
| Logout | Click on logout | User logged out | Correct | |

Table 4.1: Test plan for form authentication [adapted from Dawson (2005)].

using the Java `keytool` command. For the sake of simplicity, a default keystore password is used. Due to the use of a self-signed certificate, the browser will show an error message saying that the server's certificate is not trusted. In order to avoid this it would be necessary to sign the server certificate through an established Certificate Authority (CA). However, the use of self-signed certificates is legitime and sufficient for this dissertation project.

Since the Java Runtime Environment (JRE) does not support FOAF+SSL yet, it is necessary to bypass the standard Tomcat SSL authentication. This is done by implementing the `SSLImplementation` attribute which employs the jSSLutils library developed by Harbulot (2011). As this projects employ self-signed certificates, any kind of certificate is accepted as specified in the `acceptAnyCert` attribute. As a result, it is not necessary to go through a CA for setting up a FOAF+SSL server so that any user can deploy it.

Finally, the Tomcat configuration file `tomcat-users.xml` must be amended in order to take into account users' certificates as shown below:

```
1 <user username="CN=Sandro Cirulli" password="" roles="
    helpdesker,juniorhelpdesker"/>
2 <user username="CN=Gio Example" password="" roles="
    rddmanager,linemanager"/>
```

Listing 4.5: SSL Authentication in `tomcat-users.xml`

It is worth noticing that no password is required here but the username attribute must match the Subject field of a X509 certificate.

The deployed web application should also be configured in order to use client certificates for the authentication process. This is done by modifying the `login-config` property in the deployment descriptor `web.xml` as illustrated below. Security constraints declared for the form authentication are also valid for the client certificate authentication.

```
1 <login-config>
2   <auth-method>CLIENT-CERT</auth-method>
3   <realm-name>Resources Protected</realm-name>
4 </login-config>
```

Listing 4.6: Client certificate authentication in `web.xml`

WebID profiles for each user are generated through MyProfile (2011a). This web service creates a unique URI and generates a FOAF file that describes the user and links a client certificate to her profile. In order to use the WebID as a login credential, it is necessary to add the public key generated from the client certificate into the FOAF file as outlined in the next page.

```
<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/ns/auth/rsa#
      RSAPublicKey"/>
  <cert:identity rdf:resource="http://webid.fcns.eu/people/
      GioExample/card#me"/>
  <rsa:modulus rdf:parseType="Resource">
      <cert:hex>c426f675b[...]</cert:hex>
  </rsa:modulus>
  <rsa:public_exponent rdf:parseType="Resource">
      <cert:decimal>65537</cert:decimal>
  </rsa:public_exponent>
</rdf:Description>
```

Listing 4.7: Public key in a FOAF file

The public key is contained in the `<cert:hex>` tag and is abbreviated for the sake of simplicity. For this project, two WebIDs identifying two different users are created at the following URIs:

- `http://webid.fcns.eu/people/GioExample/card#me` for user giovanna (role linemanager).
- `http://webid.fcns.eu/people/SandroCirulli/card#me` for user sandro (role juniorhelpdesker).

Client certificates need to be imported into the browser so that they can be retrieved during the authentication process. Figure 4.8 shows client certificates successfully imported into Google Chrome 12. Each browser has a different procedure to access advance settings and to import client certificates. A strong password has been used for protecting both certificates and it is important to notice that this is the only password which the user needs to remember. In order to add extra security, it is possible during the import to tick the option forcing the use of a password every time a client certificate is requested. Although this defeats the purpose of not using passwords, it may be considered as a complementary approach for added security.

### 4.3.2 *Walk-through*

The code for the access control system deployed with FOAF+SSL is included in the companion CD as NetBeans project `webid`[5]. This time the system runs on port 8443 to ensure secured and encrypted transfer of information. The browser may block the access to the server as the site's certificate is not trusted. As explained above, this is due to the fact that the server's certificate is self-signed and not verified by a CA. By clicking on "Proceed anyway" the self-signed certificate is added to the list of exceptions in the browser settings.

5 Notwithstanding the name of the NetBeans project, the folder name in the CD is JSPExamples(J2EE1.4). In fact, the NetBeans project has been renamed but the folder's name could not be changed due to renaming issues in NetBeans

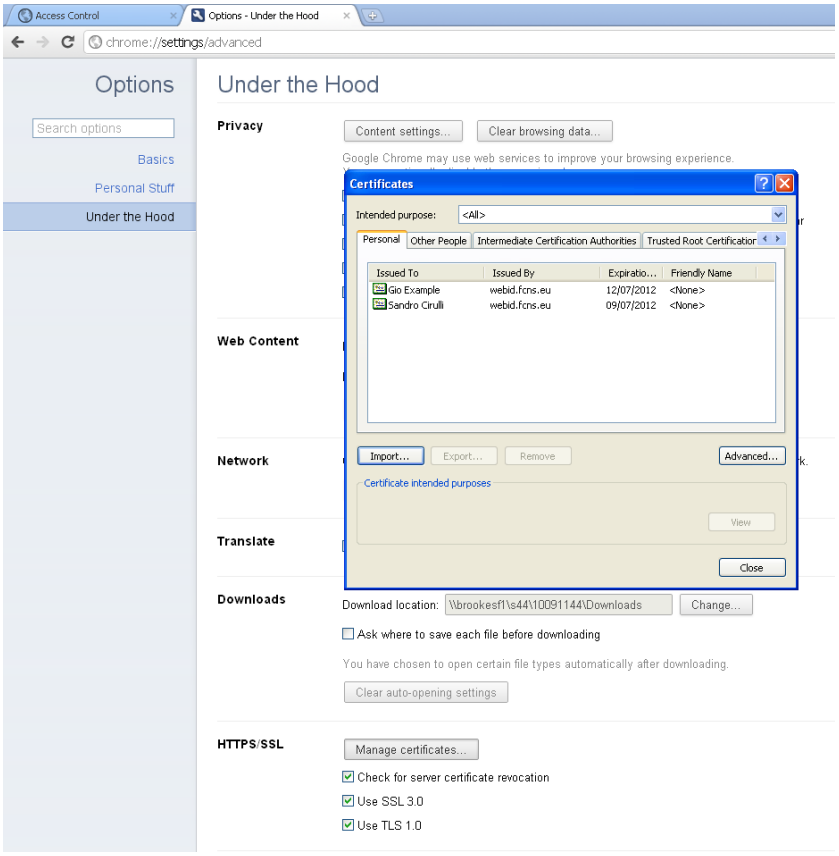


Figure 4.8: Client certificates imported in Google Chrome 12.

After clicking on the link "Login with FOAF+SSL" (figure in the access control menu 4.2) , a window pops up asking the user to select a certificate (figure 4.9). The certificate can be inspected by clicking on "View Certificate". In the certificate's details it is worth noticing the Subject field corresponding to the user name (figure 4.10) and the Subject Alternative Name field containing the WebID (figure 4.11) as these values are going to be used later.

Once the certificate is selected, the user is redirected to an external WebID authentication service at `https://foafssl.org/srv/idp` (figure 4.12. This is an identity provider for WebID allowing to authenticate a user by means of a client certificate. The service extracts the WebID from the client certificate and verifies that it matches a valid URI. Should the browser have more than one certificate in its settings, the user can choose which identity to use. After clicking on "Login to localhost" and bypassing other security screens due to the use of self-signed certificates, the server authenticates the user and sends a response as follows:

```
$relyingService?webid=$webid&ts=$timeStamp&sig=$URLSignature
```

where `$relyingService` is the landing page[6], `$webid` is the WebID of the user, `$timeStamp` is the timestamp of the response[7], and `&sig` is the encoded signature of the URL for added security. Should the user not be authenticated, an error parameter is returned in the query string and the user is redirected to the error page `error-webid.jsp`.

Figure 4.13 shows the landing page after the authentication process. The parameters in the query string are outputted on screen for clarity. The code in `index.jsp` verifies that the WebID URI exists in the ontology by calling the method `retrieveWebID()`. The alignment between an existing user in the ontology and the corresponding WebID is easily implemented through the OWL property `owl:sameAs` as illustrated below:

```
1 <foaf:Person rdf:about="http://www.sandrocirulli.net/
      dissertation/ontology/ontology.owl#giovanna">
2     <owl:sameAs rdf:resource="http://webid.fcns.eu/people/
          GioExample/card#me"/>
3     [...]
4 </foaf:Person>
```

As a result, it is possible to reuse the existing ontology, expand it with new knowledge about the user and link it to another data set.

6 In this implementation the landing page is not specified so that the server redirects by default to `index.jsp`. Indeed, the web service was slightly changed during August 2011 and the name of parameter `$relyingService` was different at the time of the software development.

7 Note that the server is probably located in United States as it returns a current time with GMT-0700.

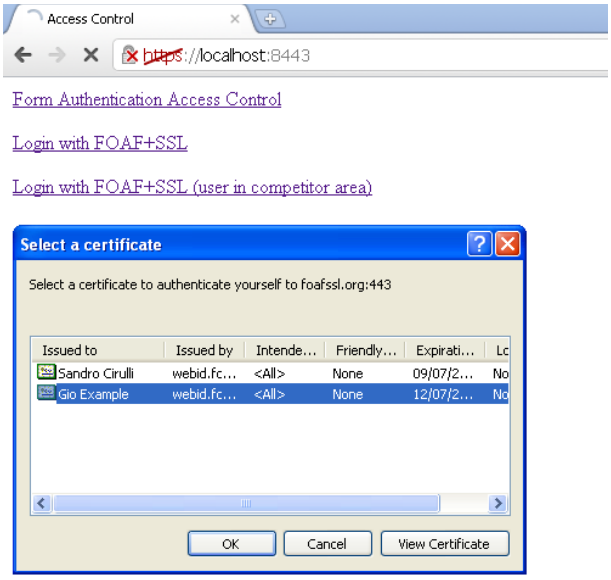


Figure 4.9: Client certificate selection.

Should the WebID exist in the ontology, the roles of the users and their permissions are retrieved in a similar manner as in the form authentication access control. However, users in the ontology are now uniquely identified by their corresponding WebID.

Access to authorized resources is shown in figure 4.14. It is worth noticing that a user is identified here through the Subject field of the client certificate - e.g. `CN=GioExample`. A similar procedure is employed for denying access as illustrated in figure 4.15.

All the steps for logging in with a WebID are described here for the sake of clarity; however, authentication with this technology would only require a single click. Furthermore, it is also possible to deploy its own authentication FOAF+SSL server instead of relying on an external service. In fact, the code is available open source and can be checked out from `https://svn.java.net/svn/sommer~svn`.

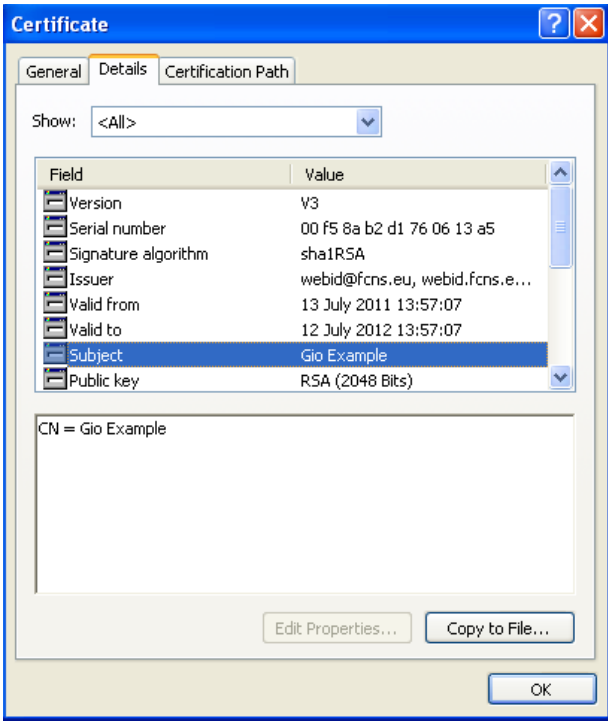


Figure 4.10: Subject in the client certificate.

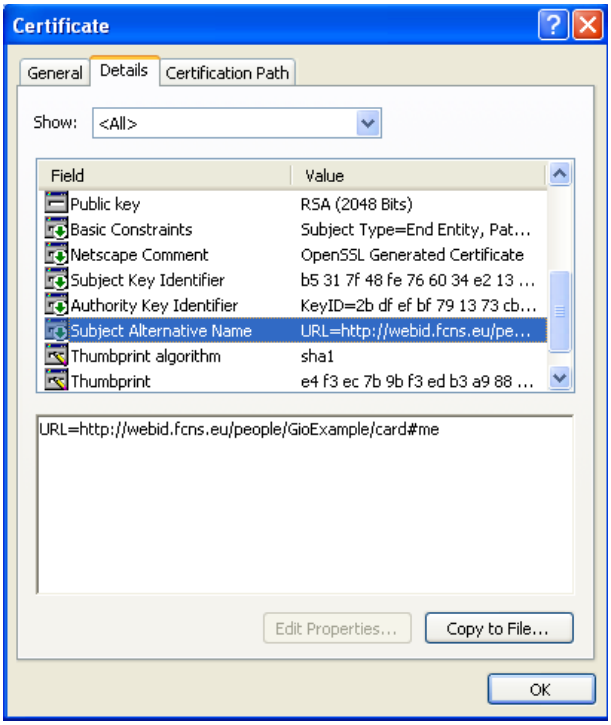


Figure 4.11: Subject alternative name in the client certificate.

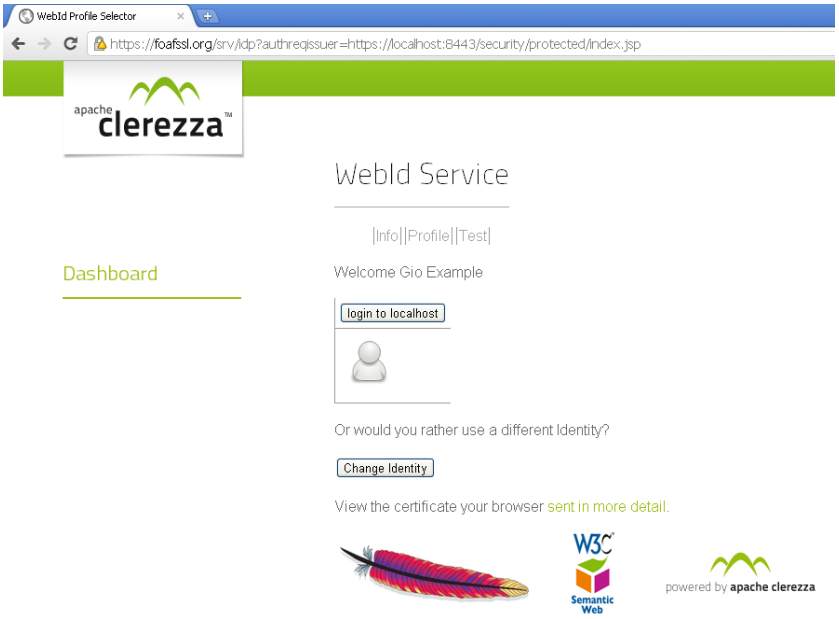


Figure 4.12: Login via foafssl.org.

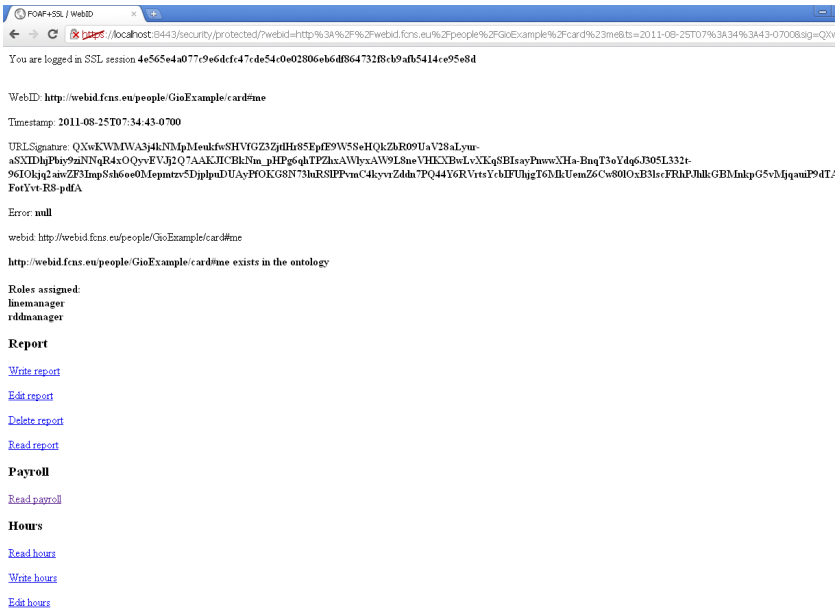


Figure 4.13: Landing page after authentication.

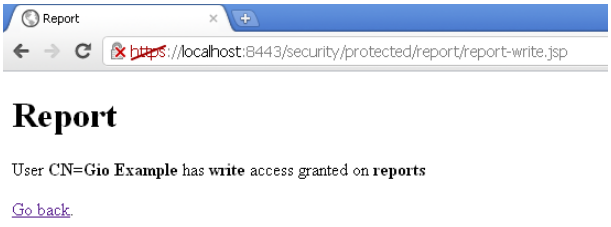


Figure 4.14: Access granted.

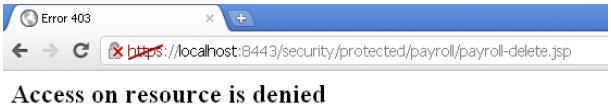


Figure 4.15: Access denied.

### 4.3.3 *Test plan*

The first test plan for the access control system via FOAF+SSL / WebID was performed in July 2011 using Google Chrome 12. However, few changes in the external authentication server occured in August and the access control system needed to be rested. The latest results are provided in the test plan in table 4.2.

## 4.4 EVALUATION

Form and client certificate authentication systems can be compared and contrasted in terms of user-friendliness, security, privacy, data decentralization and performance. The following subsections describe the advantages and drawbacks of each approach and critically evaluate them in the frame of both this project and other applications.

### 4.4.1 *User-friendliness*

One of the major shortcoming of the form authentication access control is the need for the user to remember her credentials. In fact, the presence of multiple access control systems within a corporate environment often leads to a multiplication of user names and passwords. In addition, security concerns regarding weak passwords may force the user to choose a password which is complicated to remember or to change the password every given month. As a result, the user is overwhelmed by an increasing number of business passwords; moreover, this adds up to other user names and passwords employed for personal usage so that the user tend to confuse or forget them.

| FUNCTIONALITY | TEST INPUT | EXPECTED RESULT | ACTUAL RESULT | COMMENT |
|---|---|---|---|---|
| Login linemanager | Client certificate `GioExampleNSCFEU` | Accessible resources for roles rddmanager and linemanager | Authentication server redirects to wrong page (change in authentication server in August 2011) | Amended and working correctly |
| Login juniorhelpdesker | Client certificate `GioExampleNSCFEU` | Accessible resources for role helpdesker only | Authentication server redirects to wrong page (change in authentication server in August 2011) | Amended and working correctly |
| Login other users | Client certificates for other users | Accessible resources for corresponding role(s) | Not implemented | |
| Authentication denied | No certificate | Authentication fails | Correct | |
| Authentication denied | Client certificate with no WebID | Redirects to error page | Correct | |
| Authorization denied | Manually past URL into address bar for a non authorized resource | Access denied | Correct | |
| Logout | Click on logout | User logged out | Partially correct | Require specific browser settings to force the use of client certificates at each login. Use of back button must be further tested. |

Table 4.2: Test plan for access control via FOAF+SSL [adapted from Dawson (2005)].

By contrast, the use of client certificates only requires the user to remember a single strong password for securing the certificate. Furthermore, multiple profiles can be created so that different certificates can be used for work and personal usages. However, a limitation of the client certificates is that few browsers - e.g. Firefox 3.6 - tend to render certificates in a cryptic way during the authentication process thus hindering their user-friendliness.

### 4.4.2 *Security*

Security is a major concern for access control systems based on passwords. In fact, the use of weak or default passwords is the most important cause of security breaches (Verizon, 2009). In an attempt to enforce the use of strong passwords, a security policy at Nielsen forces users to choose stronger passwords and change them every quarter. However, this has led a great number of employees to write down their passwords in an unsecured manner - e.g. store them in a plain text file or worse write them down on a piece of paper left on their desk. Since security is conflicting with user-friendliness, any attempt to enforce security policies on password may cause such chain reactions.

Conversely, authentication via FOAF+SSL offers a more secure approach since it does not involve any passwords from the client and at the same time forces the server to consistently use encrypted connections. Although the latter issue may seem obvious, at the time of writing there are still popular web services which do not use HTTPS by default when accessed from mobile devices (Mills, 2011).

However, the issue of handling authentication credentials with care becomes even more relevant for digital certificates, since if stolen they may cause identity theft. In fact, a certificate becomes the user's digital passport and should therefore be given the same consideration as passports in the real world. It is therefore crucial to educate the user to on how to use digital certificates before introducing them on a widespread basis.

### 4.4.3 *Privacy*

One of the major advantages of the WebID protocol is that allows users to keep control over their own data. In fact, the FOAF file allows the sharing of resources between users while keeping track of what is shared and with whom. FOAF files can also be extended in order to provide authorization rules between users and can automatically be aggregated by user agents and web services. Furthermore, the FOAF file can be deployed on a URI that the user owns instead of relying on an external service that may use personal data for marketing purposes (cf. Pemberton (2011). As a result, user's independence and privacy

will be increased and the user will not depend on how privacy issues are addressed by external web services. Furthermore, by exposing information via Semantic Web technologies, data can be aggregated by user agents thus enabling a social web where the owner of data is in control.

### 4.4.4 *Data decentralization*

Centralized web services force users to confine their data on a single provider or to tediously replicate the same piece of information on several websites. As a result, users tend to use the most popular services and gradually become locked in with a single service. This is particularly true for social networking websites and social web applications where personal data are kept in a silo and often used by the service provider for customized marketing. This locked-in relationship is particularly detrimental for the user for the following reasons:

- Personal data cannot be linked to other web services without the company's approval
- The user puts a great amount of effort in updating data to a web service that might close, become less popular or change terms and conditions in the future
- The user ultimately looses control and often ownership of her own data

By contrast, through the decentralized approach offered by the WebID protocol, data are not confined on a single company's server; indeed, data may be linked to other profiles and sites thus contributing to the creation of Linked Web of Trust. This also allows the web to growth in a decentralized manner rather than through a small subset of popular and privately owned websites.

### 4.4.5 *Performance*

A rough measurement of the authentication process on Tomcat shows that the form authentication is quicker than the client certificate authentication. This result has been confirmed by Hollenbach et al. (2009) who tested both authentication methods on an Apache server. This is mainly due to the use of encrypted certificates and the request of a WebID on an external service. However, after the first login both authentication methods are comparable.

## SUMMARY

This chapter has described the development of access control systems based on two different approaches, namely form authentication and FOAF+SSL. The two approaches have been discussed and compared and it has been shown that access control via FOAF+SSL offer a great number of advantages in terms of security, privacy, decentralized systems, as well as a few drawbacks.

The following chapter stems from the work undertaken to develop the access control via FOAF+SSL and extends this implementation by considering LBAC constraints.

# 5 LOCATION BASED ACCESS CONTROL

This chapter outlines the work undertaken to implement LBAC into the RBAC system developed using FOAF+SSL. It provides a walk-through of the implementation of the first LBAC scenario, strategies for the implementation of further location-based constraints, a test plan, and a discussion of the developed system and its implications in terms of technological limitations and privacy issues.

## 5.1 LBAC SCENARIOS

The code for implementing LBAC constraints extends the second access control approach based on FOAF+SSL. In particular, the implementation employs geolocation and Java servlets for restricting access to resources based on the position of the user. The LBAC scenarios described in chapter 3 are presented in form of simulations since it was not physically possible to implement and test location-based conditions due to physical and technological limitations[1].

### 5.1.1 *Configuration*

Although any web server could handle a static approach where all roles and permissions are coded in a database or an ontology at compile time, the implementation of location-based conditions requires a more dynamic approach as conditions may change at runtime. This is provided by servlets which are implementation of Java classes in a web page allowing the extension of the functionalities of a web server.

In order to run, a servlet requires a servlet container such as Tomcat and the modification of few settings in the web application's configuration file. This is done by adding the servlet name and path in the `web.xml` file as follows:

```
1 <servlet>
2     <servlet-name>CustomAuth2</servlet-name>
3     <servlet-class>CustomAuth2</servlet-class>
4 </servlet>
5 [...]
6 <servlet-mapping>
7     <servlet-name>CustomAuth2</servlet-name>
8     <url-pattern>/CustomAuth2</url-pattern>
9 </servlet-mapping>
```

Listing 5.1: Servlet configuration

1 Technological limitations are discussed in paragraph 5.3

Consequently, access to the servlet functionality is restricted to authenticated users within the web application.

### 5.1.2 *First scenario - user in competitor's office*

The implementation of the first LBAC scenario takes into account the position of a user with managerial role. In particular, it simulates the access to the system from a competitor's office using a mobile device such as a netbook or a mobile phone. The code for the first LBAC scenario is included in folder `root/NetbeansProjects/JSPExamples(J2EE1.4)/web/security/competitor` and is accessed by clicking on the link "Login with FOAF+SSL (user in competitor area)".

Since the authentication is performed via FOAF+SSL, the user is asked for a certificate and is subsequently redirected to the external WebID authentication service. Certificate `GioExampleNSCFEU` is used here to simulate access from a manager. Should the authentication be successful, the response is sent to the landing page `index.jsp` with the usual parameters and the authorization process for RBAC is performed as described in chapter 4.

An additional authorization step is necessary for validating location-based conditions and the JavaScript function `findLocation()` listed below is invoked for this purpose.

```
1 function findLocation() {
2     if (navigator.geolocation) {
3       //find location with high accuracy
4       navigator.geolocation.getCurrentPosition(
            successFunction, errorFunction,{
            enableHighAccuracy:true});
5       } else {
6       alert('Geolocation is required for this page, but
            your browser doesn&apos;t support it. Try it with
             a browser that does.');
7     }
8 }
```

Listing 5.2: JavaScript function `findLocation()`

This function adapted from Opera Software (2011a) implements the W3C geolocation Application Programming Interface (API) (W3C, 2010a) and verifies that the browser supports the retrieval of the user's position with high accuracy. All major browsers - with the notable exception of Internet Explorer 8 - support the API and ask the user to allow location detection. If the user agrees, function `successFunction(position)` is invoked in order to retrieve the current user's latitude and longitude together with additional values such as accuracy of the coordinates, altitude and its accuracy, direction of movement and speed at which the user is moving (figure 5.1). Nevertheless, retrieved values are overridden in order to simulate the

presence of a manager in a small rectangular region within a competitor's office. For this reason, the script automatically bypasses the user's consent and pops up an alert box informing the user that she is in a competitor's office (figure 5.2).

After that, the script redirects to `process.jsp` which retrieves the geolocation parameters passed and prints them out on screen. This JSP script loads the ontology[2] and adds a triple assigning to a user identified with a WebID the property `dns:hasTempRole`; this property links to a blank node containing the location, the date and the time. Consequently, a temporary role "manageronmove" is assigned to the user. Since this role is not represented in the hierarchy, it acts as a filtering mechanism for applying temporary restrictions to resources in accordance with the dynamic approach described in chapter 3. The code for creating a temporary role is outlined below:

```
1  public static void addTempRole(Model model, String user,
       String ts) {
2      // create properties
3      Property hasTempRole = model.createProperty(dns+"
           hasTempRole");
4      Property role = model.createProperty(dns+"hasTempRole")
           ;
5      Property location = model.createProperty(dns+"location"
           );
6      Property time = model.createProperty(dns+"time");
7
8      Resource manageronmove = model.createResource(dns+"
           manageronmove");
9
10     // create a blank node with properties location and
           datestamp
11     // + resource manageronmove
12     Resource userURI
13       = model.createResource(user)
14         .addProperty(hasTempRole,
15                 model.createResource()
16                     .addProperty(role, manageronmove)
17                     .addProperty(location, "
                           InCompetitorOffice")
18                     .addProperty(time, ts));
19 }
```

Listing 5.3: Temporary role added to the triple store

After that, all roles in the ontology are retrieved and corresponding permissions on resources and access types are displayed on screen (figure 5.3.) Should the user have a temporary role, servlet `CustomAuth2` is invoked in order to handle access to resources

2 In this implementation the ontology is stored locally at `root/NetbeansProjects/JSPExamples(J2EE1.4)/src/accesscontrol/ontology/ontology.owl` as new triples are added to the knowledge base and the new model needs to be saved on disk for further use.

and access types. Values in the query string are passed using the POST method so that values are not shown in the address bar; as a result, these cannot be seen by the client or manually modified. The query string includes values identifying user's role, resource, access type, and path where the resource is deployed (e.g. /CustomAuth2?user="giovanna:manageronmove:payroll:read). The code snippet below shows how the parameters are passed to the servlet CustomAuth2:

```
1  if (role.equals("manageronmove"))
2  { %>
3      <h3>Payroll</h3>
4      <!-- pass a string of the form /CustomAuth2?user="
           giovanna:manageronmove:payroll:read" to servlet
           CustomAuth2-->
5      <!-- pass control to servlet CustomAuth2 and pass query
            string using POST method -->
6      <form name="submitFormPayrollRead" method="POST" action
           ="https://localhost:8443/CustomAuth2">
7      <input type="hidden" name="username" value="<%=
           username %>"/>
8      <input type="hidden" name="role" value="<%= role %>"/>
9      <input type="hidden" name="resource" value="Payroll"/>
10     <input type="hidden" name="access" value="Read"/>
11     <input type="hidden" name="file" value="/payroll/
           payroll-read.html"/>
12     <A href="javascript:document.submitFormPayrollRead.
           submit()">Read payroll</A>
13     </form>
14     [...]
15 }
```

Listing 5.4: Parameters passed to servlet

The servlet code is available at /root/NetbeansProjects/JSPExamples(J2EE1.4)/src/CustomAuth2.java and adapts the example provided by Hunter (2001; 244–245). The servlet implements classes and interfaces from the Servlet API packages javax.servlet and javax.servlet.http. The servlet configuration requires an interaction with the ontology in order to create in memory a hash table containing a key with user name, temporary role, resource and access type as well as a value of allowed for granting access. As a result, temporary roles which may temporally override standard permissions to resources are taken into account.

After the creation of the hash table, the servlet automatically calls method doPost() to handle the POST request. Values in the request are retrieved and assembled in order to build up a string with a format similar to that in the hash table - i.e. giovanna:manageronmove:payroll:read. Subsequently, the function allowUser2() is invoked in order to verify if the access to the resource is allowed in the hash table. If so, a connection to the resource URL

is established, permissions and access types for the given user are shown and the content of the HTML file where the resource is located is printed on screen (figure 5.4). Conversely, the lack of authorization displays an error message. In order to test the correct behaviour of the servlet, `process.jsp` implements a link to a non-authorized resource leading to the error page (figure 5.5).

Finally, it is worth noticing that the connection to a URL for accessing a resource uses the HTTP protocol for the sake of simplicity. In order to use the HTTPS protocol, a self-signed certificate should be imported into the Java Development Kit (JDK) using keytool. For simulation purposes, the servlet code contains an implementation with HTTPS in the comments.

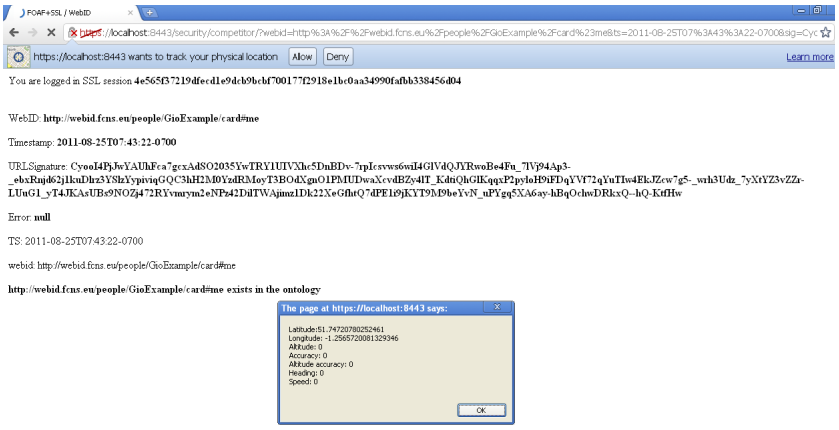


Figure 5.1: Location retrieval.

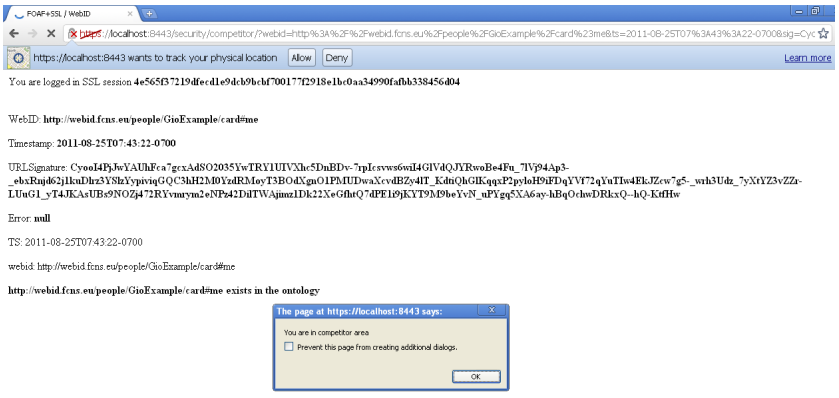


Figure 5.2: User in competitor's office.

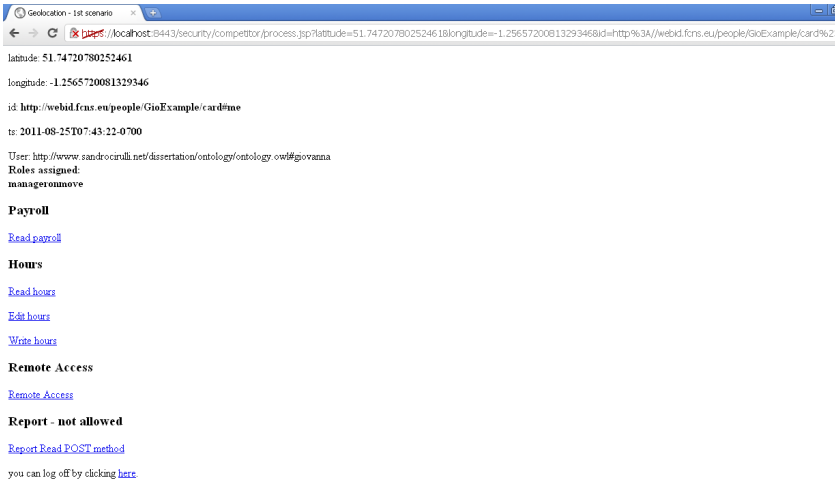


Figure 5.3: Retrieved roles, permissions and access types.

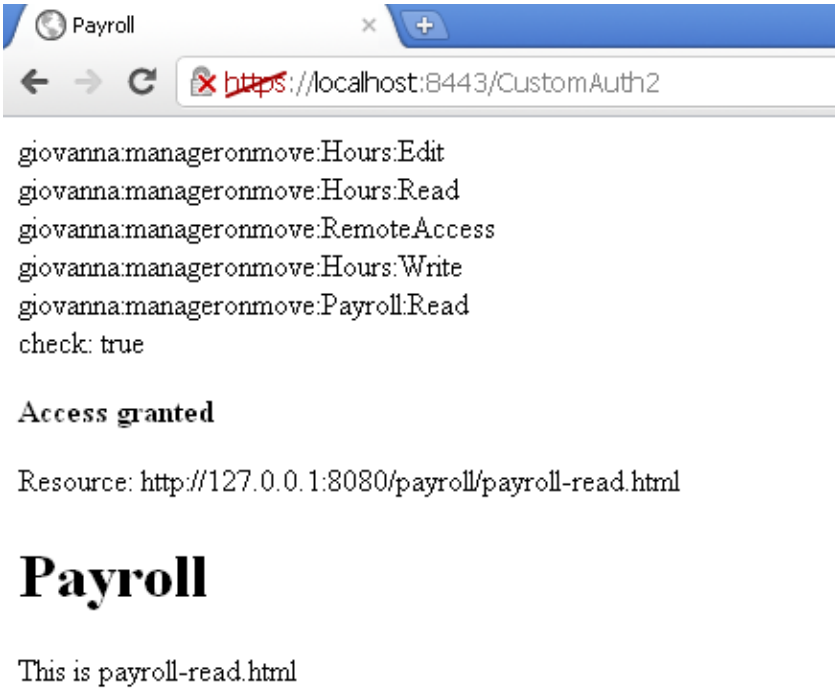


Figure 5.4: Access granted via servlet.

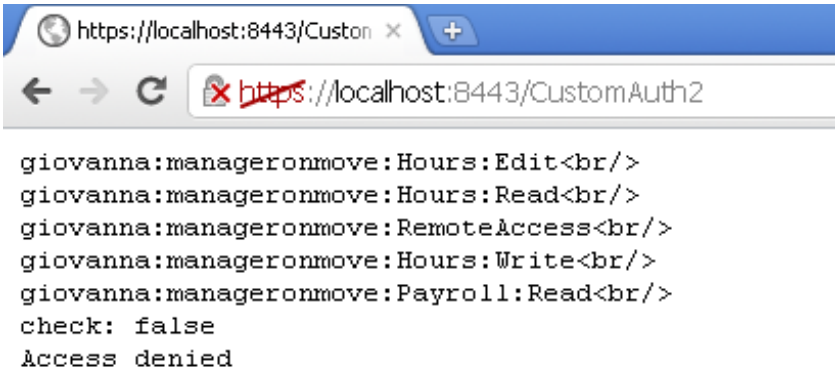


Figure 5.5: Access denied via servlet.

### 5.1.3 *Further LBAC scenarios*

Due to the lack of time, the other location-based scenarios could not be developed in this project. However, the implementation of the first LBAC scenario offers a solid base for the development of further cases. In fact, all dynamic conditions can be implemented by means of servlets thus allowing the refinement of access control policies.

The following subsections discuss how further LBAC scenarios can be implemented by combining servlets, JavaScript scripts, Jena rules and reasoning.

#### 5.1.3.1 *Detection confidence level*

According to the model presented in chapter 3, approximation in location detection is given by the detection confidence level. This value can be retrieved via the W3C geolocation API using `position.coords.accuracy` in a JavaScript script. This value expresses the approximation in meters from the coordinates retrieved. For instance, an accuracy of 0 corresponds to a confidence level of 100% whereas an accuracy of 10 denotes an approximation of 10 meters from the position retrieved. Percentage values can be assigned to the accuracy level retrieved and confidence threshold limits can be defined according to the precision required by the web application. Should the accuracy be below a given threshold limit, the position would need to be detected again. Ardagna et al. (2006; 219) have sketched a function that verifies the position up to a given number of tries and returns a boolean value in the set $\{true, false, undefined\}$. This function could be implemented within `geolocation.js` in order to account for approximation in the position detection.

#### 5.1.3.2 *Velocity*

Velocity refers to the user's speed of movement and is employed in the second location-based scenario. User's speed can be detected using `position.coords.speed` from the W3C geolocation API which is expressed in meters per second. Minimum and maximum velocities can be set using such a unit of measurement in order to define a walking speed.

When the user's position is verified in the script, an additional condition could be added in order to check that the user's speed is within the set threshold.

It is worth noticing that the W3C geolocation API can also retrieve additional information such as:

- *travel direction* expressed in degrees using `position.coords.heading`
- *position change* using `navigator.geolocation.watchPosition()` that allows the monitoring of the user's position as soon as it changes.

Such values can be integrated in the location-based conditions in order to provide a finer granularity of the user's movement.

Tracking user's position casts another interesting scenario as a change of position may trigger the removal of a permission which has already been granted. For instance, a helpdesker moving outside Nielsen's premises should not be able to access resources on the system. This can be achieved by forcing the logout functionality so that the current session is terminated and the user has to login again (for an implementation using servlets see Hunter (2001; 220)).

#### 5.1.3.3 *Density*

Density refers to the number of users within a set area and is employed in the third location-based scenario. This is the most complicated scenario to implement and requires few assumptions. Since there is no variable in the W3C geolocation API that can retrieve this information, it is necessary to involve other technologies in order to account for the position of other users. A common method used by companies for granting physical access to their premises is the use of swipe cards. For instance, a user accessing a Nielsen office with a swipe card may be recorded on a database or a triple store. Assuming that the user keeps the swipe card and other mobile devices with her, the position can be retrieved through the following techniques expressed in order of increasing granularity:

- *Door sensor* - departments within Nielsen premises may be further accessed via a swipe card so that the position in an office can be directly sent from the door sensor to the database.

- *User's device* - a swipe card may be mapped to a list of mobile devices assigned to the user so that her position can be retrieved by querying where the mobile device is currently located. This technique is particularly suited for tracking Nielsen employees as the company may provide them with such mobile devices.
- *Swipe card with GPS* - this allows the tracking of a user's exact position. This technique is particularly suited for visitors whose mobile devices cannot be known by an internal database.

A simple assertion query to a triple store of current users' positions can retrieve a boolean value stating whether the accessing user is alone within a given rectangular area. The corresponding access to resources can then be granted following the servlet implementation of the first scenario. Nevertheless, it must be noted that tracking and storing users' positions in a triple store or a database is computationally expensive and requires an ad hoc system.

#### 5.1.3.4 *Reasoning and rules*

The use of servlets may provide a full Semantic Web solution without worrying the worry of mapping users and roles in the Tomcat users' configuration file. In fact, since servlets can restricted the access to resources as shown in the first location-based scenario, it is sufficient to retrieve roles and associated permissions from the ontology and then use servlets to access the resources.

Similarly, Semantic Web rules using Jena or the Semantic Web Rule Language (SWRL)(W3C, 2004) can be employed to implement dynamic conditions. For instance, the location-based condition in the first LBAC scenario can also be expressed with the following SWRL rule:

$$inarea(?user, ?office) \wedge CompetitorOffice(?office) -> \\ hasTempRole(?user, ManagerOnMove)$$

The ability of Semantic Web technologies to align ontologies from different departments and then infer new relations also offers the opportunity to provide a distributed access control system across the whole company.

#### 5.1.3.5 *Mobile deployment*

Due to the lack of time, it was not possible to develop a widget for accessing the system via mobile devices. However, such a widget seems to be fairly straightforward to develop as the functionalities remain the same. For this purpose, the Opera Widget SDK (Opera Software, 2010) can be used as development platform since it implements the W3C geolocation API, is cross-platform, available free of charge and well documented, and also provides a mobile emulator. Since the current

web application is simple in terms of design, it is suggested that it would be fairly quick to develop a widget which complies with W3C Mobile Web Application Best Practices (W3C, 2011b) and validates with the W3C mobile validator (W3C, 2011a).

## 5.2 TEST PLAN

The LBAC system has been tested with Google Chrome 12 and is limited to the first location scenario. An initial implementation has also been tested using an iPhone in order to test the W3C geolocation API with a mobile device. Results are illustrated in the test plan in table 5.1.

## 5.3 EVALUATION

This section critically evaluates and reflects upon the solution implemented for LBAC with particular reference to the use of servlets, technological limitations in the position retrieval and privacy issues related to geolocalisation.

The use of servlets in this implementation has proved to extend the server functionalities and to account for location-based conditions using a dynamic approach. Furthermore, servlets may be employed to implement further location-based scenarios and to provide a full and distributed Semantic Web approach to access control in conjunction with ontology reasoning and rules.

Technological limitations in position retrieval have limited the implementation to a mere simulation. In fact, key parameters for determining the position of the user such as accuracy and velocity were not retrieved by the laptop or the mobile phone used during the testing phase and needed therefore to be hard-coded. It has been asserted that such values may be retrieved through GPS built-in devices in open spaces with good connectivity (The CSS Ninja, 2011b); however, the position retrieval may still be problematic with laptops and netbooks using geolocation by IP address or with mobile devices using GPS in closed spaces such as company offices. In addition, retrieving user's position with high accuracy or monitoring location changes is computationally expensive in terms of battery usage and may result infeasible when applied to all the employees of a company.

Finally, geolocalisation arises privacy issues in terms of:

- *Consent* - users must be informed that the position is about to be used by an application. The pop-up box asking for user's consent is addressing this issue.
- *Data storage* - users must be informed on how, where and how long the information is stored. This can be addressed by pro-

| FUNCTIONALITY | TEST INPUT | EXPECTED RESULT | ACTUAL RESULT | COMMENT |
|---|---|---|---|---|
| Authorization granted | Client certificate `GioExampleNSCFEU` and user in competitor's office | Access granted on payroll, hours and remote | Correct | |
| Authorization denied | Client certificate `GioExampleNSCFEU` and user in competitor's office | Access denied on report | Correct | |
| Access to resource without authentication | Manually paste correct URL with query string | Certificate asked | Resource accessed | This is due to the sending of the authorization header which is commented out in `process.jsp`. This issue is solved by importing the server certificate into the JDK and using a URL connection for the servlet on port 8443. |
| Access from a mobile device | Click on test link to retrieve position | Location parameters correctly displayed | Latitude and longitude correct, other parameters have `null` value | The mobile phone used (iPhone) and the laptop are not able to retrieve additional location parameters. |

Table 5.1: Test plan for location-based conditions [adapted from Dawson (2005)].

viding a clear privacy policy page complying with the Data Protection Act (HMSO, 1998).

- *Third-party services* - users must be informed whether the information is going to be shared with third-party services. A privacy policy on the company's intranet can also be used here.
- *Data aggregation* - accessing geolocation services from a mobile device may imply that the phone service provider or the phone manufacturer could also be able to track and store the position of the user. This information can subsequently be aggregated by third-parties without the user's consent.

The issues above have been highly debated in recent years due to widespread use of mobile devices. For a broader overview, an extensive discussion of privacy issues related to the W3C geolocation API is provided by Doty et al. (2010).

## SUMMARY

This chapter has described the integration of location-based conditions within a RBAC system developed with FOAF+SSL and discussed issues related to its implementation. The following chapter terminates the dissertation by presenting conclusions and future work.

# 6 CONCLUSION

This chapter concludes the dissertation by discussing the major difficulties encountered during the project, sketching future work and drawing the final conclusions.

## 6.1 DIFFICULTIES ENCOUNTERED DURING THE PROJECT

The main difficulties encountered during the dissertation project were related to the learning of new technologies, time management and the use of external web services.

As far as new technologies are concerned, I did not plan to use FOAF+SSL in the first place; indeed, I did not know this authentication protocol when I first planned the project. It was only at the beginning of the development phase that I came across papers regarding the protocol and thought it would be a highly suitable authentication method for an access control system based on Semantic Web technologies. Although the implementation was still in line with the objectives of my project, it caused a drift in my Gantt chart and a subsequent rescheduling of the project timeline. However, I believe that the implementation via FOAF+SSL provides a useful insight on how current advances of Semantic Web technologies may influence the way web access control systems are currently developed.

One of my major regrets is that I have not used servlets for the RBAC implementations. Indeed, this was a new technology for me and did not know exactly how and when to use servlets at the very beginning. For instance, while developing the RBAC systems I was frustrated by the fact that I could retrieve roles and permissions from the ontology but I still needed to refer back to roles coded in the Tomcat configuration for consistently accessing the resources. In fact, at the time I could not find a mechanism for denying access to a resource when a user manually enters the URL from the address bar. As the LBAC implementation has shown, servlets can bypass this issue and provide a more dynamic approach. For instance, I could have used reasoning and rules within the ontology to enforce access control policies and then deployed servlets to serve resources. This would have resulted in a better and more elegant Semantic Web solution.

Due to the learning curve for familiarising with new technologies, I ran out of time and was not able to deliver all the location-based scenarios. Indeed, I believe that my project workload was slightly too ambitious but I still wanted to implement location-based conditions as this is a very interesting problem in access control. However, I decided

to stick with my original schedule and stopped the development phase at the end of July in order to focus on the final report.

Nevertheless, I had to verify my code again when at the middle of August I realized that the external FOAF+SSL server used for the WebID authentication slightly changed its API. As a result, the redirection to the landing page was not working anymore. However, I was able to amend this fairly quickly and my implementation could then benefit from the improved user interface of the web service. At the same time, I also noticed that the ontology deployed on the web was no longer accessible. After some research, I realized that the hosting provider managing my web space was under attack and had to deactivate several hosting servers. Fortunately, the situation went back to normal after 2 days so that I could successfully retest my implementations.

The above difficulties showed me that it is not always possible to plan the full extent of risk management issues before the development phase. However, risk plans and time management charts are still good tools for tackling unexpected risks as they provides a road map for the project.

## 6.2 FUTURE WORK

This project did not fully complete the development of a Semantic Web access control system and raises few questions that still need to be properly addressed. In particular, further work is needed in the following areas:

- *Full Semantic Web solution* - this can be achieved by using inference and rules in conjunction with servlets for retrieving roles and permissions thus providing a more elegant solution.
- *Location-based scenarios* - these would need to be implemented and tested on mobile devices in order to find out how generic LBAC can be made when various location variables concurrently influence the access to resources.
- *Linked Data* - the distributed approach provided by FOAF+SSL allows the sharing of access control policies across various departments and organisations so that distributed access control can be provided.
- *ACL* - users should be allowed to modify their own access control list. An initial implementation has been provided by Hollenbach et al. (2009).
- *Performance* - FOAF+SSL authentication and authorization modules can be developed for Tomcat in a similar manner as those employed by Hollenbach et al. (2009) for the Apache web server. This would allow the calculation and possibly the improvement of the performance of FOAF+SSL on an application server.

## 6.3 CONCLUSION

This dissertation has investigated the role of Semantic Web technologies in modelling and developing a role and location based access control. The previous chapters have shown the increasing importance of Semantic Web technologies by highlighting the role that they play and the benefits that they bring in access control systems.

*Research question*

The study undertook to determine how these technologies can be used to implement role and location based access control in a real business scenario and provided practical implementations of such a system.

The results of this investigation show that access control via Semantic Web technologies offers a great number of advantages in terms of user-friendliness, security, privacy and data decentralization.

*Findings*

One of the more significant findings to emerge is that the FOAF+SSL / WebID protocol is a more suitable access control approach for Semantic Web applications thanks to its distributed approach and underlying security authentication method. Such a finding is consistent with recent studies in the field and may overcome issues related to the inherently insecure nature of passwords.

The second major finding is that role and location based conditions can be mixed to provide finer granularity in the access control system. In particular, conditions based on location are well implemented via servlets which extend the functionalities of a web server and may enforce more complex dynamic rules thus pushing the Semantic Web technologies to their full potential.

*Contribution to field*

The findings that we have identified therefore assist in the understanding of the role that Semantic Web technologies may play in shaping tomorrow's web applications as well as their ability to link data on the web in a distributed manner.

The findings of this study have a number of important implications for future practice. There is, therefore, a definite need for overcoming problems linked to passwords and providing a more secure and distributed approach to user authentication.

*Limitations of the study*

Finally, a number of important limitations need to be considered. The main caveat to note is that the developed access control system is a small though evidence-based implementation and did not attempt to account for all the issues arising from the use of Semantic Web technologies. In fact, the study was limited by the scope of this research and has only touched the surface of how Semantic Web application can be employed in access control. Further work needs to be done to assess the effects of a pure Semantic Web solution in an access control system and its implications in terms of location-based conditions, distributed data and policies, and performance.

*Future work*

# A

## OWL CODE

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-
        ns#" >
]>


<rdf:RDF xmlns="http://www.sandrocirulli.net/dissertation/
    ontology/ontology.owl#"
     xml:base="http://www.sandrocirulli.net/dissertation/
         ontology/ontology.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#
         ">
    <owl:Ontology rdf:about="http://www.sandrocirulli.net/
        dissertation/ontology/ontology.owl">
         <owl:imports rdf:resource="http://xmlns.com/foaf
             /0.1/"/>
    </owl:Ontology>



    <!--
    ///////////////////////////////////////////////////
    //
    // Annotation properties
    //
    ///////////////////////////////////////////////////
     -->

    <owl:AnnotationProperty rdf:about="&foaf;name"/>
    <owl:AnnotationProperty rdf:about="&rdfs;comment"/>



    <!--
    ///////////////////////////////////////////////////
    //
    // Datatypes
    //
    ///////////////////////////////////////////////////
```

```
45      -->
46
47
48
49
50      <!--
51      ///////////////////////////////////////////////////
52      //
53      // Object Properties
54      //
55      ///////////////////////////////////////////////////
56       -->
57
58
59
60
61      <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#grantedOn -->
62
63      <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#grantedOn">
64          <rdf:type rdf:resource="&owl;FunctionalProperty"/>
65          <rdfs:comment>corresponds to object (hasObject) in
                Finin</rdfs:comment>
66          <rdfs:range>
67              <owl:Restriction>
68                  <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#grantedOn"/>
69                  <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Resource"/>
70              </owl:Restriction>
71          </rdfs:range>
72          <rdfs:domain>
73              <owl:Restriction>
74                  <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#grantedOn"/>
75                  <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Action"/>
76              </owl:Restriction>
77          </rdfs:domain>
78      </owl:ObjectProperty>
79
80
81
82      <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#grantedTo -->
83
84      <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#grantedTo">
85          <rdf:type rdf:resource="&owl;FunctionalProperty"/>
86          <rdfs:comment>corresponds to subject (hasSubject)
                in Finin</rdfs:comment>
87          <rdfs:range>
```

```
88              <owl:Restriction>
89                  <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#grantedTo"/>
90                  <owl:allValuesFrom rdf:resource="&foaf;
                        Person"/>
91              </owl:Restriction>
92          </rdfs:range>
93          <rdfs:domain>
94              <owl:Restriction>
95                  <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#grantedTo"/>
96                  <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Action"/>
97              </owl:Restriction>
98          </rdfs:domain>
99      </owl:ObjectProperty>
100
101
102
103     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#hasActiveRole -->
104
105     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#
            hasActiveRole">
106         <rdfs:subPropertyOf rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#hasRole"/>
107     </owl:ObjectProperty>
108
109
110
111     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#hasObject -->
112
113     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#hasObject">
114         <rdf:type rdf:resource="&owl;FunctionalProperty"/>
115         <rdfs:domain>
116             <owl:Restriction>
117                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#hasObject"/>
118                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Action"/>
119             </owl:Restriction>
120         </rdfs:domain>
121         <rdfs:range>
122             <owl:Restriction>
123                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#hasObject"/>
```

```
124                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Resource"/>
125             </owl:Restriction>
126         </rdfs:range>
127     </owl:ObjectProperty>
128
129
130
131     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#hasRole -->
132
133     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#hasRole">
134         <rdfs:comment>all possible roles</rdfs:comment>
135         <rdfs:domain>
136             <owl:Restriction>
137                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#hasRole"/>
138                 <owl:allValuesFrom rdf:resource="&foaf;
                        Person"/>
139             </owl:Restriction>
140         </rdfs:domain>
141         <rdfs:range>
142             <owl:Restriction>
143                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#hasRole"/>
144                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Role"/>
145             </owl:Restriction>
146         </rdfs:range>
147     </owl:ObjectProperty>
148
149
150
151     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#hasSubject -->
152
153     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#hasSubject"
            >
154         <rdf:type rdf:resource="&owl;FunctionalProperty"/>
155         <rdfs:domain>
156             <owl:Restriction>
157                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#hasSubject"/>
158                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Action"/>
159             </owl:Restriction>
160         </rdfs:domain>
161     </owl:ObjectProperty>
162
```

```
163
164
165     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#hasTempRole -->
166
167     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#hasTempRole
            ">
168         <rdfs:subPropertyOf rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#hasRole"/>
169     </owl:ObjectProperty>
170
171
172
173     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#isSubRole -->
174
175     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#isSubRole">
176         <rdf:type rdf:resource="&owl;TransitiveProperty"/>
177         <rdfs:domain>
178             <owl:Restriction>
179                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#isSubRole"/>
180                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Role"/>
181             </owl:Restriction>
182         </rdfs:domain>
183         <rdfs:range>
184             <owl:Restriction>
185                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#isSubRole"/>
186                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Role"/>
187             </owl:Restriction>
188         </rdfs:range>
189     </owl:ObjectProperty>
190
191
192
193     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#permitted -->
194
195     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#permitted">
196         <rdfs:range>
197             <owl:Restriction>
198                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#permitted"/>
```

```
199                 <owl:someValuesFrom rdf:resource="http://
                        www.sandrocirulli.net/dissertation/
                        ontology/ontology.owl#AccessType"/>
200             </owl:Restriction>
201         </rdfs:range>
202         <rdfs:range>
203             <owl:Restriction>
204                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#permitted"/>
205                 <owl:someValuesFrom rdf:resource="http://
                        www.sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Action"/>
206             </owl:Restriction>
207         </rdfs:range>
208         <rdfs:domain>
209             <owl:Restriction>
210                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#permitted"/>
211                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Role"/>
212             </owl:Restriction>
213         </rdfs:domain>
214     </owl:ObjectProperty>
215
216
217
218     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#prohibited -->
219
220     <owl:ObjectProperty rdf:about="http://www.sandrocirulli
            .net/dissertation/ontology/ontology.owl#prohibited"
            >
221         <rdfs:range>
222             <owl:Restriction>
223                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#prohibited"/>
224                 <owl:someValuesFrom rdf:resource="http://
                        www.sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Action"/>
225             </owl:Restriction>
226         </rdfs:range>
227         <rdfs:domain>
228             <owl:Restriction>
229                 <owl:onProperty rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#prohibited"/>
230                 <owl:allValuesFrom rdf:resource="http://www
                        .sandrocirulli.net/dissertation/
                        ontology/ontology.owl#Role"/>
231             </owl:Restriction>
232         </rdfs:domain>
233     </owl:ObjectProperty>
234
```

```
235
236
237     <!-- http://xmlns.com/foaf/0.1/gender -->
238
239     <owl:ObjectProperty rdf:about="&foaf;gender">
240         <rdfs:domain rdf:resource="&foaf;Person"/>
241     </owl:ObjectProperty>
242
243
244
245     <!-- http://xmlns.com/foaf/0.1/mbox -->
246
247     <owl:ObjectProperty rdf:about="&foaf;mbox">
248         <rdfs:domain>
249             <owl:Restriction>
250                 <owl:onProperty rdf:resource="&foaf;mbox"/>
251                 <owl:allValuesFrom rdf:resource="&foaf;
                        Person"/>
252             </owl:Restriction>
253         </rdfs:domain>
254     </owl:ObjectProperty>
255
256
257
258     <!-- http://xmlns.com/foaf/0.1/mbox_sha1sum -->
259
260     <owl:ObjectProperty rdf:about="&foaf;mbox_sha1sum">
261         <rdfs:domain>
262             <owl:Restriction>
263                 <owl:onProperty rdf:resource="&foaf;mbox"/>
264                 <owl:allValuesFrom rdf:resource="&foaf;
                        Person"/>
265             </owl:Restriction>
266         </rdfs:domain>
267     </owl:ObjectProperty>
268
269
270
271     <!-- http://xmlns.com/foaf/0.1/phone -->
272
273     <owl:ObjectProperty rdf:about="&foaf;phone">
274         <rdfs:domain>
275             <owl:Restriction>
276                 <owl:onProperty rdf:resource="&foaf;mbox"/>
277                 <owl:allValuesFrom rdf:resource="&foaf;
                        Person"/>
278             </owl:Restriction>
279         </rdfs:domain>
280     </owl:ObjectProperty>
281
282
283
284     <!--
285     ///////////////////////////////////////////////
286     //
287     // Data properties
288     //
```

```
289     ///////////////////////////////////////////////////
290      -->
291
292
293
294
295     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#access -->
296
297     <owl:DatatypeProperty rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#access">
298         <rdfs:domain rdf:resource="http://www.sandrocirulli
                .net/dissertation/ontology/ontology.owl#
                AccessType"/>
299         <rdfs:range rdf:resource="&rdfs;Literal"/>
300     </owl:DatatypeProperty>
301
302
303
304     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#location -->
305
306     <owl:DatatypeProperty rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#location">
307         <rdfs:range rdf:resource="&rdfs;Literal"/>
308     </owl:DatatypeProperty>
309
310
311
312     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#time -->
313
314     <owl:DatatypeProperty rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#time">
315         <rdfs:range rdf:resource="&rdfs;Literal"/>
316     </owl:DatatypeProperty>
317
318
319
320     <!--
321     ///////////////////////////////////////////////////
322     //
323     // Classes
324     //
325     ///////////////////////////////////////////////////
326      -->
327
328
329
330
331     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#AccessType -->
332
```

```
333     <owl:Class rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#AccessType"/>
334
335
336
337     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#Action -->
338
339     <owl:Class rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#Action"/>
340
341
342
343     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#PermittedAction -->
344
345     <owl:Class rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#PermittedAction"
            >
346         <rdfs:subClassOf rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#Action"/>
347         <owl:disjointWith rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#ProhibitedAction"/>
348     </owl:Class>
349
350
351
352     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#ProhibitedAction -->
353
354     <owl:Class rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#ProhibitedAction
            ">
355         <rdfs:subClassOf rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#Action"/>
356     </owl:Class>
357
358
359
360     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#Resource -->
361
362     <owl:Class rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#Resource"/>
363
364
365
366     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#Role -->
367
368     <owl:Class rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#Role"/>
369
370
```

```
371
372     <!-- http://xmlns.com/foaf/0.1/Person -->
373
374     <owl:Class rdf:about="&foaf;Person"/>
375
376
377
378     <!--
379     ///////////////////////////////////////////////////
380     //
381     // Individuals
382     //
383     ///////////////////////////////////////////////////
384      -->
385
386
387
388     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#Hours -->
389
390     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#Hours">
391         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Resource
                "/>
392     </owl:NamedIndividual>
393
394
395
396     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#Payroll -->
397
398     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#Payroll">
399         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Resource
                "/>
400     </owl:NamedIndividual>
401
402
403
404     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#RemoteAccess -->
405
406     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#RemoteAccess">
407         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Resource
                "/>
408     </owl:NamedIndividual>
409
410
411
```

```
412     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#Report -->
413
414     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#Report">
415         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Resource
                "/>
416     </owl:NamedIndividual>
417
418
419
420     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#accountant -->
421
422     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#accountant">
423         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
424     </owl:NamedIndividual>
425
426
427
428     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#giovanna -->
429
430     <foaf:Person rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#giovanna">
431     <owl:sameAs rdf:resource="http://webid.fcns.eu/people/
            GioExample/card#me"/>
432         <rdf:type rdf:resource="&owl;NamedIndividual"/>
433         <foaf:name rdf:datatype="&xsd;string">Gio Example</
                foaf:name>
434         <hasActiveRole rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#linemanager"/>
435         <permitted>
436             <rdf:Description>
437                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#RemoteAccess"/>
438             </rdf:Description>
439         </permitted>
440         <prohibited>
441             <rdf:Description>
442                 <access rdf:datatype="&rdfs;Literal">Delete
                        </access>
443                 <access rdf:datatype="&rdfs;Literal">Edit</
                        access>
444                 <access rdf:datatype="&rdfs;Literal">Write<
                        /access>
445                 <prohibited rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Payroll"/>
446             </rdf:Description>
```

```
447         </prohibited>
448         <prohibited>
449             <rdf:Description>
450                 <access rdf:datatype="&rdfs;Literal">Delete
                        </access>
451                 <prohibited rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Hours"/>
452             </rdf:Description>
453         </prohibited>
454         <permitted>
455             <rdf:Description>
456                 <access rdf:datatype="&rdfs;Literal">Read</
                        access>
457                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Payroll"/>
458             </rdf:Description>
459         </permitted>
460         <permitted>
461             <rdf:Description>
462                 <access rdf:datatype="&rdfs;Literal">Edit</
                        access>
463                 <access rdf:datatype="&rdfs;Literal">Read</
                        access>
464                 <access rdf:datatype="&rdfs;Literal">Write<
                        /access>
465                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Hours"/>
466             </rdf:Description>
467         </permitted>
468         <permitted>
469             <rdf:Description>
470                 <access rdf:datatype="&rdfs;Literal">Delete
                        </access>
471                 <access rdf:datatype="&rdfs;Literal">Edit</
                        access>
472                 <access rdf:datatype="&rdfs;Literal">Read</
                        access>
473                 <access rdf:datatype="&rdfs;Literal">Write<
                        /access>
474                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Report"/>
475             </rdf:Description>
476         </permitted>
477     </foaf:Person>
478
479
480
481     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#hdadmin -->
482
483     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#hdadmin">
```

```
484         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
485         <isSubRole rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#
                rddmanager"/>
486     </owl:NamedIndividual>
487
488
489
490     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#helpdesker -->
491
492     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#helpdesker">
493         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
494     </owl:NamedIndividual>
495
496
497
498     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#juniorhelpdesker -->
499
500     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#juniorhelpdesker">
501         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
502         <isSubRole rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#
                helpdesker"/>
503     </owl:NamedIndividual>
504
505
506
507     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#linemanager -->
508
509     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#linemanager">
510         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
511         <isSubRole rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#
                rddmanager"/>
512     </owl:NamedIndividual>
513
514
515
516     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#manageronmove -->
517
518     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#manageronmove">
```

```
519         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
520         <permitted>
521             <rdf:Description>
522                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#RemoteAccess"/>
523             </rdf:Description>
524         </permitted>
525         <prohibited>
526             <rdf:Description>
527                 <access rdf:datatype="&rdfs;Literal">Delete
                        </access>
528                 <access rdf:datatype="&rdfs;Literal">Edit</
                        access>
529                 <access rdf:datatype="&rdfs;Literal">Write<
                        /access>
530                 <prohibited rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Payroll"/>
531             </rdf:Description>
532         </prohibited>
533         <prohibited>
534             <rdf:Description>
535                 <access rdf:datatype="&rdfs;Literal">Delete
                        </access>
536                 <prohibited rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Hours"/>
537             </rdf:Description>
538         </prohibited>
539         <permitted>
540             <rdf:Description>
541                 <access rdf:datatype="&rdfs;Literal">Read</
                        access>
542                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Payroll"/>
543             </rdf:Description>
544         </permitted>
545         <permitted>
546             <rdf:Description>
547                 <access rdf:datatype="&rdfs;Literal">Edit</
                        access>
548                 <access rdf:datatype="&rdfs;Literal">Read</
                        access>
549                 <access rdf:datatype="&rdfs;Literal">Write<
                        /access>
550                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Hours"/>
551             </rdf:Description>
552         </permitted>
553         <prohibited>
554             <rdf:Description>
555                 <access rdf:datatype="&rdfs;Literal">Delete
                        </access>
```

```
556                 <access rdf:datatype="&rdfs;Literal">Edit</
                        access>
557                 <access rdf:datatype="&rdfs;Literal">Read</
                        access>
558                 <access rdf:datatype="&rdfs;Literal">Write<
                        /access>
559                 <permitted rdf:resource="http://www.
                        sandrocirulli.net/dissertation/ontology
                        /ontology.owl#Report"/>
560             </rdf:Description>
561         </prohibited>
562     </owl:NamedIndividual>
563
564
565
566     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#rddmanager -->
567
568     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#rddmanager">
569         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
570         <rdfs:comment>a person who manages people</
                rdfs:comment>
571         <isSubRole rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#
                manageronmove"/>
572     </owl:NamedIndividual>
573
574
575
576     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#sandro -->
577
578     <foaf:Person rdf:about="http://www.sandrocirulli.net/
            dissertation/ontology/ontology.owl#sandro">
579     <owl:sameAs rdf:resource="http://webid.fcns.eu/people/
            SandroCirulli/card#me"/>
580         <rdf:type rdf:resource="&owl;NamedIndividual"/>
581         <foaf:name rdf:datatype="&xsd;string">Sandro
                Cirulli</foaf:name>
582         <permitted rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Hours"/>
583         <permitted rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#
                RemoteAccess"/>
584         <prohibited rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Report"/
                >
585         <hasActiveRole rdf:resource="http://www.
                sandrocirulli.net/dissertation/ontology/
                ontology.owl#juniorhelpdesker"/>
586     </foaf:Person>
587
588
589
```

```
590     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#seniorhelpdesker -->
591
592     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#seniorhelpdesker">
593         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
594         <isSubRole rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#
                helpdesker"/>
595     </owl:NamedIndividual>
596
597
598
599     <!-- http://www.sandrocirulli.net/dissertation/ontology
            /ontology.owl#sysadmin -->
600
601     <owl:NamedIndividual rdf:about="http://www.
            sandrocirulli.net/dissertation/ontology/ontology.
            owl#sysadmin">
602         <rdf:type rdf:resource="http://www.sandrocirulli.
                net/dissertation/ontology/ontology.owl#Role"/>
603     </owl:NamedIndividual>
604 </rdf:RDF>
```

Listing A.1: OWL code

# B

# RISK MANAGEMENT

| **Risk Event** | **Likelihood** | **Impact** | **Detection Difficulty** | **When** |
|---|---|---|---|---|
| Hardware and software failure | 1 | 5 | 1 | Any time |
| Non-compatible technologies | 1 | 4 | 1 | Deployment |
| Requirements drifting | 3 | 3 | 3 | Implementation |
| Excessive prototypes iterations | 2 | 3 | 2 | Design |
| Non-responsive client | 3 | 2 | 2 | Any time |
| Study and other commitments | 4 | 4 | 3 | Any time (likely in May) |

Table B.1: Risk assessment form [adapted from Gray & LarsonGray & Larson (2008; p. 203)].

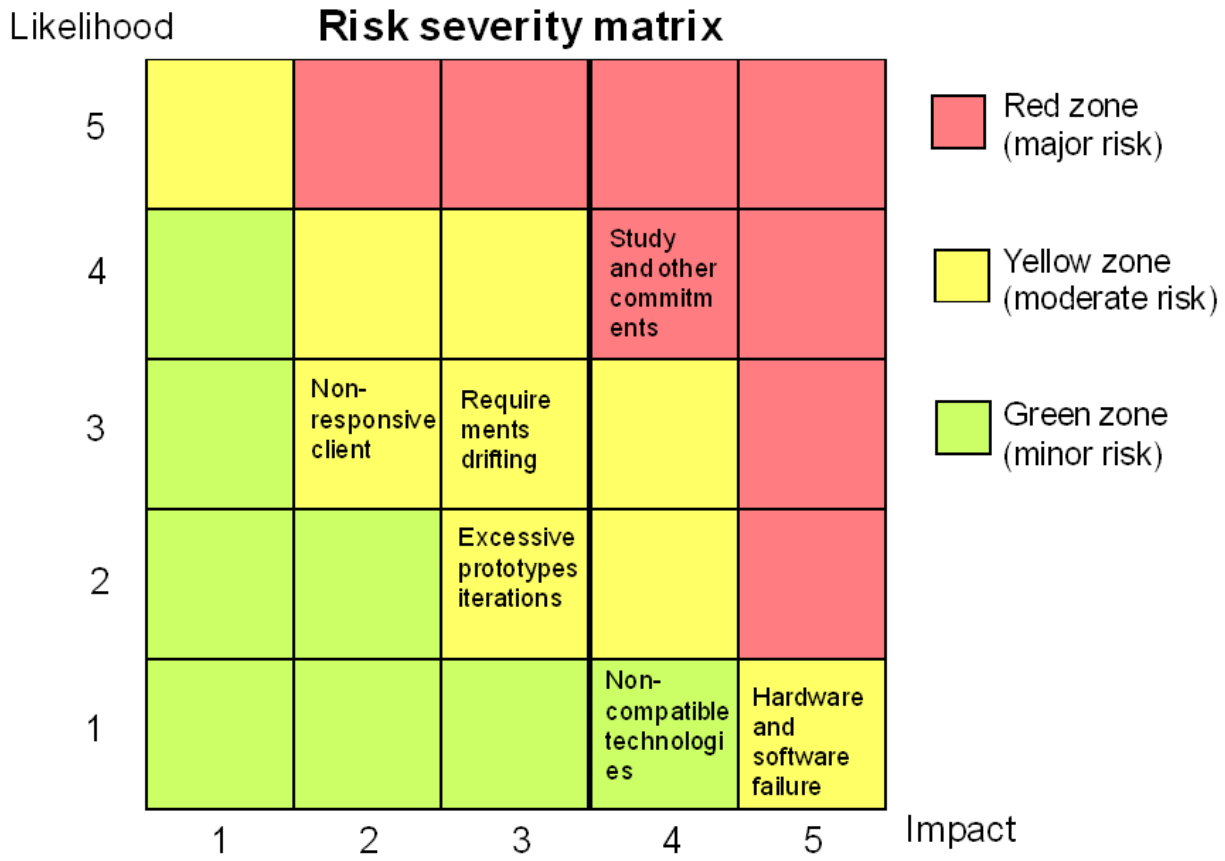


Figure B.1: Risk severity matrix [adapted from Gray & Larson (2008; p. 204)].

| Risk Event | Response | Contingency Plan | Trigger | Responsible |
|---|---|---|---|---|
| Hardware and software failure | Mitigate: keep regular and multiple backups | Rewrite dissertation and software | Files lost | Sandro |
| Non-compatible technologies | Avoid: use W3C standard technologies | Change to compatible technology or change hardware | Not able to deploy | Sandro |
| Requirements drifting | Mitigate: prototypes validation | Explain drifting in the report | Design and implementation don't comply with prototypes | Sandro |
| Excessive prototypes iterations | Avoid: limit to 2 iterations | Set new maximum number of iterations | End of 2nd iteration | Sandro / AC Nielsen manager |
| Non-responsive client | Mitigate: organise regular meetings with client | Proceed without client's feedback | No feedback after contact | Sandro / AC Nielsen manager and helpdeskers |
| Study and other commitments | Mitigate: plan reduced work on dissertation in May and plan commitments according to Gantt chart | Ask for an extension | Drift from Gantt chart | Sandro / Fareena |

Table B.2: Risk response matrix [adapted from Gray & Larson (2008; p. 209)].

# C

# COMPANION CD

The attached CD contains the electronic version of this dissertation and the code of the software development. It is structured in the following folders:

A. report
   a) dissertation-10091144.pdf
B. NetBeansProjects
   a) form-auth
   b) JSPExamples(J2EE1.4)
C. apache-tomcat-6.0.32
D. certificates
E. JenaLibraries

## C.1 REFERENCES

Adida, B. (2011). *An unwarranted bashing of Twitter's OAuth*. Available at: http://benlog.com/articles/2010/09/02/an-unwarranted-bashing-of-twitters-oauth (Accessed 2 September 2011).

ANSI (2004). *American National Standard for Information Technology - Role Based Access Control*. ANSI INCITS 359-2004.

Apache Software Foundation (2011a). *Apache License - Version 2.0*. Available at: http://www.apache.org/licenses/LICENSE-2.0 (Accessed 2 September 2011).

Apache Software Foundation (2011b). *Apache Tomcat*. Available at: http://tomcat.apache.org (Accessed 2 September 2011).

Apache Software Foundation (2011c). *Apache Tomcat 6.0 - SSL Configuration HOW-TO*. Available at: http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html (Accessed 2 September 2011).

Ardagna, C. A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., & Samarati, P. (2006). Supporting location-based conditions in access control policies. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, ASIACCS '06, (pp. 212–222)., New York, NY, USA. ACM.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*, 34–43.

Bringhurst, R. (2002). *The Elements of Typographic Style*. Point Roberts, WA: Hartley & Marks Publishers.

Brittain, J. & Darwin, I. F. (2003). *Tomcat: the definitive guide*. Sebastopol; Farnham: O'Reilly.

Cirio, L., Cruz, I. F., & Tamassia, R. (2007). A role and attribute based access control system using semantic web technologies. In *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems - Volume Part II*, OTM'07, (pp. 1256–1266)., Berlin, Heidelberg. Springer-Verlag.

Dale, F. A. (2002). *A customisable web portal with Perl and XML*. MSc Thesis. Oxford Brookes University.

Dawson, C. W. (2005). *Projects in computing and information systems: a student's guide*. Harlow: Prentice Hall.

Doty, N., Mulligan, D. K., & Wilde, E. (2010). Privacy Issues of the W3C Geolocation API, Available at: http://escholarship.org/uc/item/0rp834wf (Accessed 2 September 2011).

Dubray, J. (2011). *Is OAuth 2.0 Bad for the Web?* Available at: http://www.infoq.com/news/2010/09/oauth2-bad-for-web (Accessed 2 September 2011).

Ferraiolo, D. & Kuhn, R. (1992). Role-Based Access Control. In *15th NIST-NCSC National Computer Security Conference*, (pp. 554–563).

Ferraiolo, D. F., Kuhn, D. R., & Chandramouli, R. (2003). *Role-Based Access Control*. Norwood, MA, USA: Artech House, Inc.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* PhD thesis, University of California, Irvine.

Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W., & Thuraisingham, B. (2008). ROWLBAC: representing role based access control in OWL. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, SACMAT '08, (pp. 73–82)., New York, NY, USA. ACM.

FOAF Project (2011). *The Friend of a Friend (FOAF) project.* Available at: http://www.foaf-project.org/ (Accessed 2 September 2011).

Gray, C. F. & Larson, E. W. (2008). *Project Management: The Managerial Process.* (4th ed.). New York: McGraw-Hill.

Harbulot, B. (2011). *jSSLutils - Java SSL Utilities.* Available at: http://code.google.com/p/jsslutils/ (Accessed 2 September 2011).

HMSO (1998). *Data Protection Act.* Available at: http://www.legislation.gov.uk/ukpga/1998/29/contents (Accessed 2 September 2011).

Hollenbach, J., Presbrey, J., & Berners-Lee, T. (2009). Using RDF metadata to enable access control on the social semantic web. In *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2009)*, (ISWC 2009), Washington, DC.

Hunter, J. (2001). *Java Servlet Programming*. Sebastopol; Farnham: O'Reilly.

Li, N., Byun, J.-W., & Bertino, E. (2007). A Critique of the ANSI Standard on Role-Based Access Control. *IEEE Security and Privacy*, *5*, 41–49.

Mills, E. (2011). *Facebook seeking encryption for apps, mobile.* Available at: http://news.cnet.com/8301-27080_3-20036172-245.html (Accessed 2 September 2011).

MyProfile (2011a). *MyProfile Demo Page.* Available at: http://webid.fcns.eu/webidgen.php (Accessed 2 September 2011).

MyProfile (2011b). *MyProfile Manifesto.* Available at: http://myprofile-project.org/manifesto.php (Accessed 2 September 2011).

Nielsen (2011). *The Nielsen Company.* Available at: http://www.nielsen.com (Accessed 2 September 2011).

OAuth (2011). *OAuth Community Site.* Available at: http://oauth.net/ (Accessed 2 September 2011).

OpenID Foundation (2011). *OpenID Foundation website.* Available at: http://openid.net (Accessed 2 September 2011).

Opera Software (2010). *Opera Widgets SDK.* Available at: http://dev.opera.com/sdk/ (Accessed 2 September 2011).

Opera Software (2011a). *How to use the W3C Geolocation API.* Available at: http://dev.opera.com/articles/view/how-to-use-the-w3c-geolocation-api (Accessed 2 September 2011).

Pemberton, S. (2011). *Why you should have a Web Site.* Available at: http://www.w3.org/2008/Talks/1022-Steven_Pemberton/ (Accessed 2 September 2011).

Power, D., Slaymaker, M., & Simpson, A. (2009). On Formalizing and Normalizing Role-Based Access Control Systems. *Comput. J.*, *52*, 305–325.

Stanford Center for Biomedical Informatics Research (2011a). *Choosing between versions of Protégé.* Available at: http://protegewiki.stanford.edu/wiki/Protege4Migration (Accessed 2 September 2011).

Stanford Center for Biomedical Informatics Research (2011b). *The Protégé Ontology Editor and Knowledge Acquisition System.* Available at: http://protege.stanford.edu/ (Accessed 2 September 2011).

Story, H., Harbulot, B., Jacobi, I., & Jones, M. (2009). FOAF+SSL: RESTful authentication for the social web. In *European Semantic Web Conference*, Workshop: SPOT2009, Heraklion, Greece.

Story, H. (2011a). *FOAF+SSL: a first implementation.* Available at: http://blogs.oracle.com/bblfish/entry/foaf_ssl_a_first_implementation (Accessed 2 September 2011).

Story, H. (2011b). *How to setup Tomcat as a FOAFSSL server.* Available at: http://blogs.oracle.com/bblfish/entry/how_to_setup_tomcat_as (Accessed 2 September 2011).

The CSS Ninja (2011b). *Accessing the GPS in iPhone Safari.* Available at: http://www.w3.org/Submission/SWRL (Accessed 2 September 2011).

The OpenSSL Project (2011). *OpenSSL: The Open Source Toolkit for SSL/TLS.* Available at: http://www.openssl.org (Accessed 2 September 2011).

Verizon (2009). *2009 Data Breaches Investigations Report*. Available at: http://www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf (Accessed 2 September 2011).

W3C (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Available at: http://www.w3.org/Submission/SWRL (Accessed 2 September 2011).

W3C (2010a). *Geolocation API Specification.* Available at: http://www.w3.org/TR/geolocation-API (Accessed 2 September 2011).

W3C (2011a). *FOAF+SSL+OpenID*. Available at: http://www.w3.org/wiki/FOAF%2BSSL%2BOpenID (Accessed 2 September 2011).

W3C (2011b). *Mobile Web Application Best Practices Cards.* Available at: http://www.w3.org/2010/09/MWABP/ (Accessed 2 September 2011).

W3C (2011). *OWL Web Ontology Language.* Available at: http://www.w3.org/TR/owl-ref/ (Accessed 2 September 2011).

W3C (2011a). *W3C mobileOK Checker.* Available at: http://validator.w3.org/mobile (Accessed 2 September 2011).

W3C (2011b). *W3C Semantic Web Activity*. Available at: http://www.w3.org/2001/sw (Accessed 2 September 2011).

W3C (2011c). *Web Access Control.* Available at: http://www.w3.org/wiki/WebAccessControl (Accessed 2 September 2011).

W3C (2011d). *Web Access Control / Vocabulary.* Available at: http://www.w3.org/wiki/WebAccessControl/Vocabulary (Accessed 2 September 2011).

W3C (2011e). *WebIDs and the WebID Protocol*. Available at: http://www.w3.org/wiki/WebID (Accessed 2 September 2011).

W3C (2011f). *What does FOAF+SSL give you that OpenID does not* . Available at: http://blogs.oracle.com/bblfish/entry/what_does_foaf_ssl_give (Accessed 2 September 2011).

## C.2 UNCITED REFERENCES

Hughes, B. & Cotterell, M. (2009). *Software Project Management.* (5th ed.). London: McGraw-Hill Higher Education.

Maciaszek, L. (2007). *Requirements analysis and system design* (3rd ed.). Harlow: Addison-Wesley.

Sommerville, I. (2007). *Software Engineering* (8th ed.). Harlow: Addison-Wesley.

Wikipedia (2011a). *Access control*. Available at: http://en.wikipedia.org/wiki/Access_control (Accessed 2 September 2011).

Wikipedia (2011b). *Authentication*. Available at: http://en.wikipedia.org/wiki/Authentication (Accessed 2 September 2011).

Wikipedia (2011c). *OAuth*. Available at: http://en.wikipedia.org/wiki/Oauth (Accessed 2 September 2011).

## COLOPHON

This dissertation was typeset with LaTeX 2ε using Kile on Ubuntu 10.04 and MiKTeX on Brookes computers equipped with Windows XP. The typographic style was taken from André Miede's work available for LaTeX via CTAN as "`classicthesis`" under GNU GPL licence. André Miede's work is in turn inspired by Bringhurst's "*The Elements of Typographic Style*" (Bringhurst, 2002).

*Final Version* as of September 2, 2011 at 0:07.