Data security in location-aware applications: an approach based on RBAC

M.L. Damiani*

Dipartimento di Informatica e Comunicazione,
Università degli Studi di Milano,
via Comelico 39/41, 20135 Milano, Italy
EPFL-IC-LBD, Lausanne, CH
E-mail: damiani@dico.unimi.it
*Corresponding author

E. Bertino

Department of Computer Sciences,
Purdue University,
West Lafayette, IN 47907, USA
E-mail: bertino@cerias.purdue.edu

P. Perlasca

Dipartimento di Informatica e Comunicazione,
Università degli Studi di Milano,
via Comelico 39/41,
20135 Milano, Italy
E-mail: perlasca@dico.unimi.it

Abstract: Data security in a mobile context is a critical issue. Over the last few years a new category of location-based services, the Enterprise LBS (ELBS), has emerged focusing on the demands of mobility in organisations. These applications pose challenging requirements, including the need of selective access to ELBS based on the position of mobile users and spatially bounded organisational roles. To deal with these requirements a novel access control system, named GEO-RBAC, has been developed. GEO-RBAC extends the NIST RBAC (Role-Based Access Control) standard with the notions of spatial role, role-dependent position, role schema and role instance. Further, roles become enabled/disabled based on the position of the user. In the paper we present GEO-RBAC, a full-fledged RBAC-based model, consisting, like RBAC, of three distinct components: the Core GEO-RBAC, the Hierarchical GEO-RBAC and the Constrained GEO-RBAC. The paper focuses on the innovative aspects that have been introduced in the model to account for the spatial dimension. Further, a rigorous specification of the model (reference model) is presented.

Keywords: access control; data security; geographic information systems; information and computer security; location aware applications; mobile systems; RBAC; role based access control.

Reference to this paper should be made as follows: Damiani, M.L., Bertino, E. and Perlasca, P. (2007) 'Data security in location-aware applications: an approach based on RBAC', *Int. J. Information and Computer Security*, Vol. 1, No. 1/2, pp.5–38.

Biographical notes: Maria Luisa Damiani is currently working as an Assistant Professor in the Department of Computer Science and Communication at the University of Milan. She has been a Researcher at public and private research laboratories in Italy for several years, before joining the University of Milan. Her current research specifically focuses on geographical data security and privacy, spatial data warehousing and applications in geology. She is currently involved in the European Project GeoPKDD as subcontractor of EPFL.

Elisa Bertino is a Professor of CS and ECE at Purdue University and a Research Director at the Centre for Education and Research for Information Assurance and Security (CERIAS). Previously, she was a CS Professor and Chair of the Department of Computer Science and Communication at the University of Milan. Her main research interests include security, database systems, object technology, multimedia systems, and web-based information systems. She co-authored the book *Intelligent Database Systems* (Addison Wesley). Elisa Bertino is a Fellow Member of IEEE and a Member of ACM and has received the 2002 Technical Achievement Award by IEEE Computer Society. She has served as Programme Committee Member of several international conferences, such as ACM SIGMOD, VLDB, ACM OOPSLA and is currently serving as a Co-editor-in-Chief of the *VLDB Journal*.

Paolo Perlasca received his PhD in Computer Science at the University of Milan. He is an Assistant Professor at the Department of Computer Science and Communication of the University of Milan. His research interests are in the areas of access control, database security, integrity and protection.

1 Introduction

The rapid evolution in the technologies for wireless communication and position determination has pushed the demand for location-aware applications and in particular Location-Based Services (LBS). LBS are *services that integrate a mobile device's location with other information so as to provide added value to the user* (Spiekermann, 2004). A popular LBS is the E-911 service, which provides emergency responders with the location of the phone caller. Over the past few years, in addition to the LBS for the consumer market, a new category of services, the Enterprise LBS (ELBS), has emerged focusing on the demands of mobility in organisations. ELBS are becoming increasingly relevant and a significant and rapid growth of the market is estimated in the near future. Among the enterprises that deploy ELBS, we include, however, not only the companies operating in the field and using ELBS for mobile resource management and tracking but also the various communities whose members, because of their functional role in the organisation, need to access the common information resources through LBS. Healthcare and leisure organisations belong to this category as well as military organisations and civilian coalitions created in response to a crisis, e.g. natural disasters.

Despite the increasing interest in ELBS, little attention has been generally paid to understand the security requirements that the organisational dimension introduces with respect to LBS applications, which is what we focus on here. To introduce the problem we first present a possible usage scenario: consider an LBS application for a university campus. The campus consists of various buildings and areas, such as departments, libraries and recreative areas, each occupying a position in space. From an organisational point of view, people in the campus can play one or more functions, such as teachers, students, visitors and, say, library subscribers. The members of the campus are connected to a wireless network and equipped with a location-aware PDA by using which they access various LBS services. Since the PDA displays the services that are available at each instant, the user can select what service to request. John is a student and also a library subscriber. When located within one of the libraries of the campus, John is presented with various services, not necessarily location-based, e.g. the *book-loan* service to request the loan of a book. As John leaves the library such services are no longer available until he entered some library again. John, however, as a student can also request the class-timetable from whatever position in the campus extent, because the service is provided only to the students in the campus. This scenario allows us to point out some important requirements:

- In enterprises the mobile members are characterised not only by an identity but also by an organisational function that is a *role*. Broadly speaking, a role is a set of rights and duties denoted by a name and assigned to a subject who plays that role (Moffett and Lupu, 1999). Depending on the roles, the information resources that the individuals need may vary and thus also the services to be provided by the LBS application. It seems thus reasonable to devise different services based on the requester's roles. This calls for the development of access control mechanisms supporting the specification of which user can access which service in which context, based also on the user's role.
- The services that the user can request at a given instant of time depend both on the position and on the user's role. In the running example, John, when located within the library, can request the loan of a book because subscriber of the library.
  It seems thus important to associate a spatial context with each role in order to define where the role of the individual is recognised and thus which services can be invoked in that position. Further, it seems natural to define such a context in terms of a *semantic space*, that is a space, which besides a geometric extent has also a meaning, like the library in the example.
- The space in which the mobile user is embedded is bounded. Space is bounded for a variety of reasons, because of application-dependent constraints, for example, the boundaries of the campus; because of technological constraints, such as the network extent; because of marketing reasons (i.e. the broader the area covered by the service, more the user shall pay) or because of security reasons as such in the case of services in a military base. It follows that the spatial context of a role *r* is limited as well; thus the user is recognised to play role *r* only if located inside the spatial context. It is thus necessary to account for the fact that the roles may be effective or not depending on the user's position.
- Spatial roles should to be defined not only at different semantic details but also over extents that have different sizes. In the running example, the campus represents the spatial extent of the role *CampusMember*, while the library is the extent of the role *LibrarySubscriber*. Further, since a user can play more than one role, it seems important to prevent a user from playing roles that are conflicting over the same or interrelated space, like the roles of student and teacher at the same department.

To deal with these requirements and in particular to regulate the access to services based on the position of users in a bounded space, we present a comprehensive access control model called *GEO-RBAC*, based on the RBAC model (Role-Based Access Control Model) (Ferraiolo et al. 2001). In the past few years, the RBAC model has gained general acceptance as the leading model for access control in networked applications. Furthermore, it has been recently approved as a standard by the American National Standards Institute (ANSI) and a number of organisations are today applying this standard in specialised domains. Nevertheless, the utilisation of RBAC for the controlled access to spatial resources, especially in innovative applications such as LBS and mobile applications, has become a research theme only in recent times.

The model we present results from the extension of the GEO-RBAC model reported in Bertino et al., (2005). The basic concept underlying GEO-RBAC is that of *spatial role*, that is a role which is assigned a spatial context, defined by a region in space. In addition, besides denoting a set of permissions as in the classical RBAC, a spatial role is characterised by a status, which is *enabled* or *disabled* depending on the position of the user. As the user moves across different spatial contexts, the status of roles may change as well and thus also the permissions which may be granted to the user because of the enabled roles. A permission corresponds to a service. Thus saying that a permission *prm* is granted to a role *r* means that the users playing role *r* are authorised to access the service corresponding to *prm*. Ultimately, the set of services which are accessible to a user may vary in time, depending on the user's position.

The GEO-RBAC, like RBAC, include three distinct components:

- The *Core GEO-RBAC* embodies the essential aspects of the model and in particular the notion of spatial roles.
- The *Hierarchical GEO-RBAC* extends the model with the notion of hierarchies of spatial roles. As a novel contribution, the model introduces the distinction between *replaceable* and *non-replaceable* role. A replaceable role is a role which can be replaced by a 'less powerful' role in the hierarchy.
- The *Constrained GEO-RBAC* introduces static and dynamic separation of duty relations to prevent the user from playing two or more conflicting spatial roles, possibly in interrelated regions.

In the paper we provide a complete definition of the components of the GEO-RBAC model and point out relevant issues in their development and related solutions. To the best of our knowledge, this is the first complete definition of a spatial RBAC model encompassing not only the core concepts but also the hierarchies and constraints.

The rest of the paper is organised as follows. In the Section 2, we introduce some preliminaries concepts, in particular the RBAC model and the spatial model adopted in GEO-RBAC to represent the position of the user and the spatial characteristics of the role. We then present the three components of the model, first informally to enable a simpler comprehension of the basic concepts and afterwards in formal terms in order to provide a detailed and rigorous description of the *Reference Model*. Sections 6 and 7 concern the related work and concluding remarks, respectively. In appendix, in order to combine theory and practice and provide the reader with a running specification of the model, we report the XML specification of the core component.

2 Background

2.1 RBAC

The RBAC model, as defined in the NIST standard, consists of four basic sets of elements: *users*, *roles*, *permissions and sessions*.

- *User* is as a human being or an autonomous agent.
- *Role* represents the function of a user within a community. The community can be a structured organisation, for example, a business enterprise, or a more informal community, e.g. the citizens of a city. A role confers a set of permissions on the user.
- *Permission* is an approval to perform an operation on one or more objects. An object is a resource that shall be protected. An operation is an executable image of a program, which upon invocation executes some function for the user over some object. The types of operations and objects depend on the application context in which RBAC is deployed. For example, in a file system, operations might include read, write and execute; in a database management system, operations might include insert, delete, append and update.
- *Session*, When the user logs in, a session is established during which the user activates some subset of roles that he or she is assigned. The permissions available to the user of the session are thus the permissions assigned to the roles that are currently active across all the users sessions.

Over the above sets of elements, a number of relations are defined. The *User-Assignment* (*UA*) relates users to roles through a many-to-many relationship; a user can therefore be assigned multiple roles and the same role assigned to different users. The *Permission-Assignment* (*PA*) relation relates roles and permissions again through a many-to-many relationship; thus a role can be assigned multiple permissions and similarly each permission can be assigned to multiple roles. The function *SessionUser* maps each session into a user, whereas the *SessionRole* function maps a session onto a set of roles, namely, the roles that are active in the session. The above sets and relations as a whole constitute the basic RBAC layer, the *Core RBAC* defined as follows:

Definition 1 (Core RBAC): *Core RBAC consists of the following components:*

- *The sets U*, *R*, *PRMS and SES represent the set of users*, *roles*, *permissions and sessions. We define:*
  - $UA \subseteq U \times R$. *The user-assignment relation that assigns users to roles.*
  - *Assigned User:* $R \rightarrow 2^{U}$. *The mapping from a role to a set of users.*
  - $PA \subseteq R \times PRMS$. *The permission-assignment relation that assigns permissions to roles.*
  - *Prms Assignment:* $R \rightarrow 2^{PRMS}$. *The mapping of a role into a set of permissions.*
  - *Session User:* $SES \rightarrow U$. *The mapping from a session to a user.*
  - *Session Role:* $SES \rightarrow 2^{R}$. *The mapping from a session to a set of roles.*

Core RBAC defines the minimum collection of RBAC elements, element sets and relations for a RBAC system to be defined. Besides the Core RBAC, the model consists of additional components that can optionally be used: *Hierarchical RBAC* and *Constrained RBAC*. The Hierarchical RBAC component adds relations for supporting role hierarchies. A hierarchy is a partial order defining a seniority relation between roles, whereby senior roles acquire the permissions of their juniors. Constrained RBAC adds Separation of Duty (SoD) constraints to the RBAC (Ferraiolo et al., 2001). SoD constraints are in general used to prevent conflicts of interest in an organisation possibly arising because of the usage of conflicting roles. Roles are conflicting if a user can gain the authorisation from them to exercise conflicting operations. Two different SoD constraints are considered: the static and the dynamic one. Static constraints prevent the assignment of conflicting roles to users, so that a user assigned to a given role can never be assigned to a role conflicting with the first one; in the dynamic case, the constraints enforce the activation of conflicting roles within a user session so that a user cannot have conflicting roles active at the same time.

2.2 The spatial data model

The spatial data model we adopt is instrumental in representing the position of the user and the spatial properties of roles. In compliance with current geo-spatial standards (ISO/TC211, 2003), we describe the spatial objects that are located in the reference space (*SPACE*), in terms of *simple features* (ISO/TC211, 2003) (hereinafter, features). Features have an identity, thematic properties and can be mapped onto a position in SPACE. We denote a feature through its identifier. For example, *UniMi* is the identifier of the spatial object that describes the properties of a campus and its position. The position of a feature is represented through a *geometry*, which can be of type point, line or polygon or recursively a collection of geometries. Further, geometries can be related by different types of relationship. Among them, the reference set of topological relations is {*Disjoint*, *Touch*, *In*, *Contains*, *Equal*, *Cross*, *Overlap*}. These relations are binary, mutually exclusive (if one is true, the others are false), and they are a refinement of the well-known set of topological relations proposed by Clementini, Di Felice and van Oosterom (1993). To exemplify, the *Contains*(x, y) relationship between geometries x and y holds when all points of y are also points of x. Moreover, features have an application-dependent semantics that is expressed through the concept of *feature type*. A feature type captures the intentional meaning of the entity. Examples of feature types: *Campus*, *Department* and *Library*. The *extension* of a feature type is a set of semantically homogeneous features. For example, the extension of the feature type *Library* is the set {*CentralLibrary*, *ScientificLib*, *Humanities*}.

3 Overview of the components

In this section we focus on the distinguishing and innovative concepts of the model, especially with respect to RBAC whilst the full definition of the model is presented in Section 4. This section is subdivided into three subsections, one for each component of the model.

3.1 Core-GEO-RBAC

By adapting the definition in Ferraiolo et al. (2001) to our model, we say that Core GEO-RBAC defines *a minimum collection of elements which are required to completely achieve a GEO-RBAC system*. The basic notion is that of spatial role, which describes a spatially bounded function for a user. Two other concepts, however, are noteworthy: (a) the *position model*, which describes the position of the mobile user; (b) the notions of *role schema* and *role instance*, which provide a representation of the role at two levels: the intensional and the extensional level. We now introduce each of these concepts.

3.1.1 The position model

The notion of position is fundamental, since it is one of the characterising aspects of the user. Unlike most proposals in mobile computing that describe the position uniquely in geometric terms, e.g. a point, we introduce, for sake of flexibility, the distinction between the *real* and the *logical* position. The real position corresponds to the position of the user on Earth acquired through some positioning technology. Real positions can thus be represented as geometric elements of different types since, depending on the chosen technology and accuracy requirements, they may correspond to points or polygons. Conversely, the *logical position* is defined at a higher level of abstraction to represent positions in a way that is almost independent from the underlying positioning technology. Further, besides a geometry, the logical position has a semantics. For example, logical positions can be a house, an address number or a road which are represented in terms of *spatial feature types*. The logical position is computed from real positions by using a *location mapping function*. For example, a location mapping function can be defined to map a position acquired through GPS onto the closer road segment. More formally, given a feature type *ft*, a *position mapping function* for *ft* is a m_{ft} that, given a real position *rp*, returns a logical position corresponding to an instance of *ft* having *rp* as real position. As we will see, since the localisation may respond to different application requirements, for example, with respect to accuracy, the meaning of location may vary depending on the role. For example, the position of a generic campus member may be coarsely defined in terms of campus sectors, assuming that the campus area is subdivided into sectors, whereas the position of the teacher can be represented at a higher resolution by an address.

3.1.2 Spatial role

The concept of *spatial role* is the distinguishing aspect of our model. To account for the spatial context, a *spatial role* is defined not only by a name as in RBAC but also by a *role extent*. The extent of a role defines the boundaries of the region contained in the reference space. A user, who has been conferred a role *r*, is thus recognised to effectively play such a role only when logically located within the extent of *r*. For example, *CampusMember* (*UniMi*) is a spatial role: *CampusMember* is the role name and *UniMi* is the role extent, specifically the identifier of a spatial feature of type *Campus* denoting, among the others, a polygonal space. An individual, who is a member of the campus, is recognised to play the role of *CampusMember* and thus authorised to invoke the services associated to the role, only when located in the polygonal extent of the campus.

Each role is assigned a set of *permissions*. A permission corresponds to a service. For example, the service *BookLoan* of the running example is assigned to the spatial role *LibarySubscriber(MyLibrary).* The terms *permission* and *service* are used in this paper in an interchangeable way. Like RBAC, when a user connects to the LBS application, a new *session* is started. Then the user selects the roles, among those which have been assigned to him/her that he/she wishes to play. This set represents the *session roles*, and the elements of the set are said to be *activated*. Unlike RBAC, however, in our model session roles have a status in addition, which is *enabled* or *disabled*. For a session role to be *enabled*, the user should be logically located within the space of the corresponding role extent. As the user moves, the status of roles change. Therefore, depending on the position of the user, only a subset of the session roles is enabled and permissions granted. As a result the set of services which the user can access at a given point in time depends both on the session roles and on the position of the user.

3.1.3 Role schema and instance

To provide a more compact representation of semantically homogeneous roles defined over different extents, we introduce the distinction between role schema and role instance. A *role instance* is a role defined over a specific extent, in compliance with the role schema. Note that the terms *roles instance* and *spatial role* are used as synonymous. A *role schema* defines some common properties of roles with a similar meaning. Specifically, a role schema defines: (a) a common name for a set of roles; (b) the type of role extent; (c) the type of logical location; (d) the mapping function relating the real position with the logical position. For example; *CampusMember* (*Campus, Sector,* m_{sector}) is a schema, the instances of which are roles having the following properties: the roles have the same name *CampusMember* and the logical position of the users is identified by a sector of the campus, assuming that the campus is subdivided in zones, which is computed by applying function m_{sector} to the real position. Once a schema is specified, the corresponding instances can be simply created by specifying the role name and its extent, for example, *CampusMember(UniMi)*.

Because roles are assigned permissions and because of the two different levels of role representation, it seems reasonable to assign permissions to both role schemas and role instances. The permissions which are assigned to a schema are then inherited and shared by all the instances of the schema. For example, if we assign the service *getMap* to the role schema *CampusMember*, it means that such a service can be accessed by the members of all campus. For the sake of flexibility, however, permissions can also be assigned to single role instances.

3.1.4 Example

To summarise the characterising aspects of spatial roles, we present an example which shows how schemas and instances are specified.

1 Consider first the set FT of spatial feature types representing the spatial objects of interest. For brevity we omit the specification of the instances.

FT = {*Campus*, *Department*, *Sector*, *Library*, *Address*}

2 Role schemas are defined for the members of the campus, the students, the teachers and the library subscribers. The CampusMember schema specifies that members are

recognised as such when located in a sector of the campus while students and teachers are recognised when located in some buildings of a department. Similarly, the subscriber of a library is recognised when in a library.

RoleSchemas = {*CampusMember*(*Campus*, *Sector*, m_{sector}), *Student*(*Dept*, *Building*, $n_{building}$), *Teacher*(*Dept*, *Building*, $n_{building}$), *LibrarySubscriber*(*Library*, *Address*, $k_{address}$)}

3 Role instances are defined for the previous role schemas. Only one instance of the *CampusMember* schema is specified because we assume a unique campus. Two instances of the student schema are defined for two departments. Similarly, two roles of the LibrarySubscriber schema are defined for two different libraries as follows:

RoleInstances = {*CampusMember*(*UniMi*), *Teacher*(*Dept*#1), *Teacher*(*Dept*#2), (*Student*(*Dept*#1), *Student*(*Dept*#2), *LibrarySubscriber*(*MyLib*), *LibrarySubscriber*(*OtherLib*)}

4 Permissions, thus services, are assigned to role schemas and instances. The service *book-loan* of the running example is assigned to the schema of *LibrarySubscriber* and is thus inherited by all instances of the given schema. Instead, the service *class-timetable* is assigned to both the schema *Teacher* and *Student*. Assume also that *the get-map* is a service assigned to the CampusMember schema to provide a service of map-based guidance.

3.2 Hierarchical GEO-RBAC

Besides Core GEO-RBAC, Hierarchical GEO-RBAC (HGEO-RBAC) has been specified as an additional component of the model to support spatial role hierarchies and thus simplify both the specification and the management of roles as well as their permissions. Spatial role hierarchies are defined by introducing a partial order ≤ over the set of roles. If we say that *CampusMember*(*UniMi*) ≤ *Student*(*Dept*#1), we mean that the role *CampusMember* in the campus *UniMi* is *less powerful* than the role *Student* performed at the Computer Science Department in the same campus. By *less powerful* we mean that the user has fewer permissions and thus can access fewer services. The introduction of hierarchies poses a number of issues. The first is whether the notion of hierarchy is applied only to role instances or role schemas or both. The second issue concerns the need of re-defining the concept of enabled roles. The two issues are discussed in the following.

3.2.1 Role hierarchies

For the sake of generality, hierarchies are defined at both extensional and intensional level. The first level corresponds to the role instance level whilst the second level to the role schema. On one hand, the schema hierarchy simplifies the specification and management of roles, since the permissions assigned to schemas are inherited by all the descendants in the hierarchy and therefore do not have to be explicitly defined for each distinct schema. On the other hand, the instance hierarchy concerns the roles directly conferred to the user. Because of the peculiar characteristics of the instance hierarchy, here we consider only this aspect. To simplify the introduction of the concept of hierarchy, we first describe it through examples.

Consider the two roles r_1 = *CampusMember*(*UniMi*) and r_2 = *Student*(*Dept*#1). If we state that $r_1 \leq r_2$, it means that:

- An individual who is a student at *Dept*#1 is also a member of the campus *UniMi*. Therefore, the permissions which are assigned to a campus member are also assigned to a student.
- The role extent of r_2 is spatially contained in the role extent of r_1, that is, the spatial extent of *Dept*#1 is contained in the spatial extent of *UniMi*.
- If role r_2 is enabled, r_1 is also enabled. Moreover, the logical position of the user in the role of student is spatially contained in the logical position computed for the same user in the role of campus member.

We represent the hierarchy by extending the notion of *role graph* (Nyanchama and Osbornm, 1999). In particular, we use the role graph to represent the set of roles through a lattice in which the nodes represent the roles and the edges represent the precedence relationship. In addition to the user-defined roles, every role graph has a *MaxRole* ($\top$) and a *MinRole* ($\perp$). *MaxRole* has assigned the union of all permissions. Further, because the role is spatial, it has an extent also. Such extent results from the intersection of all extents and can be empty. *MaxRole* is introduced to ensure that the precedence relationship is always defined; however, it is most likely the case that, in order to improve security through SoD, no user is assigned the permission to use this role. *MinRole* represents the minimum set of privileges available to all roles, possibly empty. Like *MaxRole*, *MinRole* has an extent. In this case the extent is the whole reference space, that is *SPACE*.

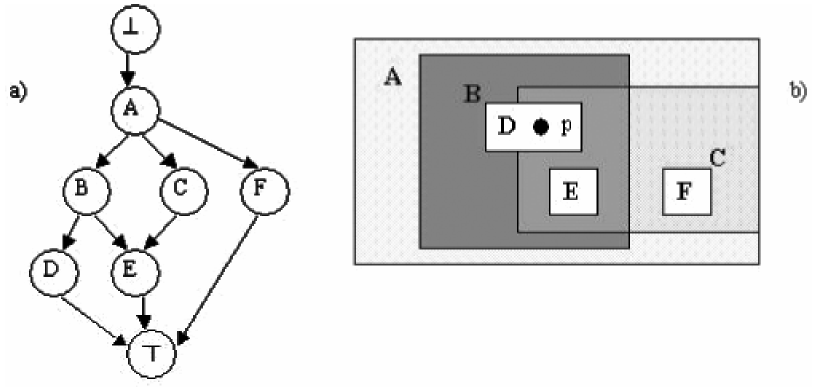
We draw the role graph without redundant edges through a Hasse diagram. In the paper we use the convention that the *MinRole* is drawn at the top. A role preceding *r* is an *ancestor* of *r*. To illustrate first the syntactical aspects of the hierarchy, consider the set of generic roles R = {$A(s_0)$, $B(s_1)$, $C(s_2)$, $D(s_3)$, $E(s_4)$, $F(s_5)$} where *X*(*Ext*) denotes the role *X* with a spatial extent identified by *Ext*. Without loosing in generality, we assume that roles are univocally identified by their names. Assume $A(s_0) \leq B(s_1)$; $A(s_0) \leq C(s_2)$; $A(s_0) \leq F(s_5)$; $B(s_1) \leq D(s_3)$; $B(s_1) \leq E(s_4)$ and $C(s_2) \leq E(s_4)$. The corresponding graph is reported in Figure 1a.

For the sake of readability, in the graphical representation, the nodes of the graph are labelled only with roles names. An arrow from *X* to *Y* means $X \leq Y$. We assume the role extents in Figure 1b. Note that the extents of two non-comparable roles, those roles which are not the one ancestor of the other, such as roles *B* and *C*, can overlap. Therefore, if a user is located in the intersection area of two roles, both of them can be enabled. Moreover, it should be noticed that the containment relationship between extents is not a sufficient condition for the roles to be comparable, since the role ordering is application-dependent. In the example, the extent of role *F* is contained in that of *C*, but the roles are not comparable.

3.2.2 Semantics of enabled roles

Because of the spatial context, the introduction of the instance hierarchy introduces a new issue. The problem can be formulated as follows: *given a set of session roles S and given a point p* $\in$ *SPACE, which are the roles in S which are enabled when the user of the session is in position p.* With reference to the role graph in Figure 1, assume that the set *S* of session roles consists of the roles $S = \{D,E\}$. Which roles are enabled when the user is in position p?

Figure 1 (a) Role hierarchy and (b) role extents.



We claim that there is no unique answer. In the example, we can devise two interpretations: (a) the first one is that the set of enabled roles is $ER = \{D,B,A\}$. The motivation is that p is contained in only one of the two session role extents, in particular in the extent of D; therefore, ER contains D as well as its ancestors because of the definition of hierarchy, that is B and A; (b) the second interpretation is that $ER = \{D,C,B,A\}$, which differs from the previous one because of element C. The motivation is that p is contained not only in the extent of D but also in the extent of C. Because C is an ancestor of E, which is a session role, it seems reasonable to include it in the set of enabled roles. The intuition behind this second interpretation is that the user not only can play the roles that he has selected but also a weaker version of them, represented by their ancestors in the role hierarchy.

The above example shows that, in order to determine the set of enabled roles, we have to define what happens if a role is *not enabled*. As we have seen, there are two possible interpretations. Which of them is the most suitable depends on the semantics of roles and thus on the requirements of the applications. Therefore, for the sake of generality, we propose a model in which the behaviour to adopt is assumed to be specific for each role and explicitly defined. When the user is allowed to play, in place of the session role r, a weaker role, in the sense discussed above, we say that r is a *replaceable role* (R-role); otherwise, we say that r is a *non-replaceable* role (NR-role). We call this property of the role as *replacement property*.

3.2.3 A model for R-roles and NR-roles

To represent the replacement property of a role r, the basic idea is to introduce an attribute *dist* (distance) indicating the maximum distance which is allowed for an ancestor to replace the role. Specifically, $dist = 0$ means that the role cannot be replaced by any other role and thus it is a NR-role; if $dist > 0$, then the role can be replaced by a role at the maximum distance *dist* and therefore it is a R-role. By restricting the value of the distance attribute, we can introduce the constraint that a role cannot be replaced by a role that is *too far*, thus too generic. This results in a greater expressivity.

We now discuss more specifically how the property is specified. The *replacement property* is described both in the role schema and in the role instance definition, in order to characterise the single instance. The role schema is thus extended with the *dist* attribute, which defines whether a role is a R-role or a NR-role. All the instances sharing the same schema inherit the same distance value. The general form of the schema for role *role* is as follows:

$Role(ext, loc, m_{loc}, dist)$

where $dist \in N$ is the *distance* that indicates whether the role is replaceable or not. An example is the following. Consider a hierarchy stating that the schema *Student* is more powerful than the immediate ancestor, the schema *CampusMember*. We state that any instance of the role *Student* can be replaced by an instance of *CampusMember* by specifying the following schema:

Student(*Department*, *Building*, *m*, 1).

The value 1 assigned to the *distance* property means that an instance of *Student* can be replaced by an instance of some immediate predecessor in the hierarchy, that is, in the example, by an instance of *CampusMember*.

To enhance flexibility, the value of the property can, however, be specified also for single instances. The structure of the role instance is thus extended with an attribute labelled *dist*, which, if not NULL, overrides the value of the corresponding attribute defined at schema level. The value defined at schema level is thus the *default* value for all the instances. An example is the role instance:

Student(*Dept*#1, 2).

The value 2 of the distance property overrides the value defined at schema level. The specification of the property at instance level is useful to introduce exceptions.

An important remark to be done is that the introduction of the notion of replaceable role arguably affects the algorithm, which determines the set of enabled roles in a session, since the location of the user must be confronted not only with session roles, to determine which of these are enabled, but eventually also with the ancestors of session roles which are R-roles.

3.3 The constrained GEO-RBAC

The third component of the model is the Constrained GEO-RBAC, which adds SoD relations to the GEO-RBAC model. SoD relations are used to enforce conflict of interest policies that the organisations may employ to prevent the users from exceeding a reasonable level of authority for their positions (Ferraiolo et al., 2001). Conflicts of interest, in particular, arise as a result of the simultaneous assignment of two mutual exclusive permissions or roles to the same subject. Following the NIST standard, we focus specifically on the constraints specifying the roles to be mutually disjoint. A generic constraint is thus expressed as (*role_set, n*), where *n* is the number of roles which are mutually exclusive in *role_set*. The precise meaning of role set depends, however, on whether the SoD is *static* or *dynamic*. A *Static Separation of Duty* (SSD) constraint means that a user cannot *be assigned n* roles from the given role set. Conversely, a *Dynamic Separation of Duty* (DSD) constraint means that a user cannot select, that is activate, *n* roles in a session from the given role set. To deal with the spatial context, however, it seems important to extend the conventional classification of constraints to account for the specific characteristics of our model. Two orthogonal criteria, therefore, have been introduced that allow us to characterise constraints based on whether they are: (a) expressed at instance or schema level; (b) based on spatial criteria or not. By combining these two criteria, we obtain four classes of constraints, one of which is not meaningful. We discuss first the two criteria and then introduce the proposed classification of constraints.

1 *Instance vs. schema-based constraints*. At instance level, the constraint is expressed by enumerating the roles which are requested to be disjoint. The form of the constraint is $(\{r_1, \ldots r_n\}, n)$ with $n > 1$ where each r_i, $i = 1, \ldots n$, is a role instance. For example, the constraint ({*Teacher*(*Dept*#1), *Student*(*Dept*#1)}, 2) states that if a user is assigned the role of teacher in department *Dept*#1, he/she cannot be assigned the role of student in the same department and vice versa. It can be noticed that the members of the role set not only are requested to be known but the set of roles should also be modified; because of the addition and removal of elements, it may occur that the constraints are to be updated as well.

   If the constraint is defined at schema level, the members of the role set are specified intensionally in terms of role schemas. We say that two roles schemas rs_1, rs_2 are mutually disjoint when all the pairs of elements $< r_1, r_2 >$ with r_1 and r_2 instances, respectively, of rs_1 and rs_2 are disjoint. The example (*Teacher*, *Student*, 2) states that a teacher cannot be at the same time a student in any campus and in any department within a campus. A constraint defined at schema level enables a more compact representation of generalised constraints over sets of roles, that otherwise would require a lengthy specification. Further, the constraint is not affected by the operations that modify the set of role instances.

2 *Non-spatial vs. spatial constraints*. Non-spatial constraints are those which are close to the standard RBAC constraints, since they do not consider the spatial dimension of roles. The spatial component characterises instead the constraints that are spatial. In particular, a constraint is spatial when the roles that are mutually exclusive are those which fulfil a given spatial relationships. The intuition is that conflicts of interest may arise not only because of the semantics of roles but also because the roles are defined over conflicting extents. It is the case, for example, of roles which cannot be played over the same region.

By combining these criteria we obtain four classes of constraints: non-spatial constraints at instance and schema level; spatial constraints at instance level and at schema level. As we will see, one of these four classes is not meaningful. We characterise each class as follows:

1 *Non-spatial constraint at instance level* . The purpose of the constraint expressed as (*role_instance_set*, *n*) is to prevent the user from playing *n* or more role instances among those specified in the *role_instance_set*. For example, we can state that a teacher in campus *A* cannot be a member of campus *B* and campus *C*.
   Such constraint is expressed as: ({*Teacher*(*A*), *CampusMember*(*B*), *CampusMember*(*C*)}, 3). This constraint constitutes the primitive constraint at the lowest level of abstraction. It is very similar to the standard SoD constraint in RBAC.

2 *Non-spatial constraint at schema level*. The purpose of the constraint expressed as (*role_schema_set*, *n*) is to prevent the user from playing *n* distinct role instances from *n* schemas in *role_schema_set*. An example is the constraint (*CampusTeacher*, *CampusStudent*, 2), which states that an individual cannot be a teacher and a student at the same time independent of the campus, thus whatever role instance the user plays from the given schemas. The constraint is thus applied to all the possible pairs of instances. Notice that this condition could also be formulated by introducing a distinct constraint for each pair of instances of teacher and student. The operation, however, would result into a large number of constraints expensive to update.

A particular case occurs when the schemas in the *role_schema_set* are not distinct. Consider the constraint (*CampusMember*, *CampusMember*, 2). Based on the previous definition, the meaning of the constraint is that an individual cannot play two distinct roles of the schema *CampusMember*, that is she cannot be a member of two different campus. This particular case is interesting since it seems reasonable in practice to constrain an individual to play a given role in a unique or, however, limited number of spatial extents. In order to abbreviate the expression, we express the constraint as (*role_schema_n*). The meaning is that a user cannot play *n* with *n* > 1 role instances of the same schema.

3 *Spatial constraint at schema level*. The idea is to prevent the user from playing role instances of two different schemas if the role extents fulfil a specified spatial relationship. The motivation is that because roles are defined over spatial extents, it may be the case that conflicts of interest arise when roles are played over nearby regions. Without loosing in generality, we assume that the constraint is expressed between the pairs of role schemas as follows: ($role\_schema_1$, $role\_schema_2$, *spatial_rel*) where *spatial_rel* is the spatial relationship which holds between instances of $role\_schema_1$ and $role\_schema_2$. For example, the constraint (*Teacher*, *Student*, *Overlaps*) means that an individual cannot simultaneously be a teacher and a student over overlapping regions (while in disjoint regions, that is campus it is allowed).

Notice that in case of role hierarchies, we have to specify as additional requirement that the role schemas $role\_schema_1$ and $role\_schema_2$ should not be comparable, because the hierarchy already introduces a constraint of spatial containment over the extents of instances along a path and thus this additional constraint could be redundant or inconsistent with the definition of hierarchy. We observe also that a spatial constraint defined over role instances, like *(Teacher(A)*, *Student(B)*, *2)* has little sense since the spatial relationship between the role extents *A* and *B* is predefined and thus a spatial constraint would be meaningless.

4 The reference model

We now introduce a rigorous definition of the model. The purpose is to provide a comprehensive definition of the components, thus including all the aspects of the model.

4.1 The core GEO-RBAC

Core GEO-RBAC is presented as organised in a number of logical parts, one for each major set of the RBAC model, that is roles, permissions, users, sessions. The general structure of the model is illustrated in Figure 2. We use the graphical representation adopted for RBAC. In defining the model, we refer to the notation introduced in the previous section and summarised in Table 1.

Figure 2 Core GEO-RBAC

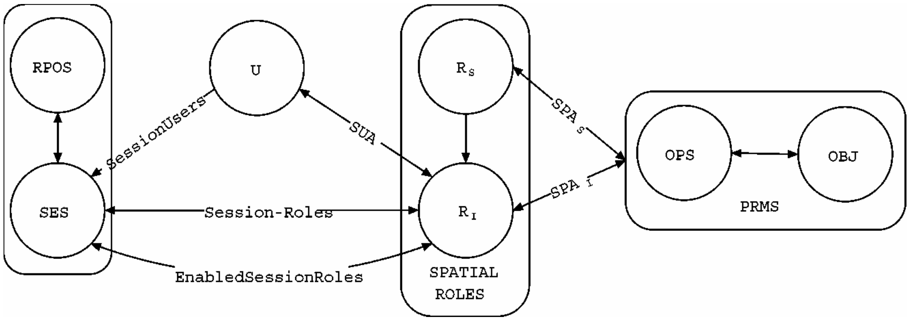


Table 1 Notation for the main sets used in GEO-RBAC

Notation	*Meaning*
FT	Feature types
F	Features
R	Role names
SES	Sessions
U	Users
REXT_FT	Role extent types
LPOS_FT	Logical positions types
LPOS	Logical positions
RPOS	Real positions
M	Position mapping functions
OPS	Operations
OBJ	Objects

Preliminarily, we introduce the following function and relation.

- *The position of a feature*. Given a feature f, the function $LocObj(f)$ returns a geometry if f is a spatial feature or undefined ($\perp$) otherwise.
- *Partial ordering of feature types*. Let FT be the set of feature types and $ft_i, ft_j \in FT$, with $i \neq j$, be two elements of the set. We say that ft_i is contained in ft_j , denoted by $ft_i \subseteq_{ft} ft_j$, if for each feature f_i of type ft_i, a feature f_j of type ft_j exists such that the geometry of f_i is contained in the geometry of f_j . For example the relation of containment between *Town* and *Region* is written as *Town* $\subseteq_{ft}$ *Region*. As we will see, such relationship will be useful in characterising the relationships between locations and roles.

4.2 Role schema and instance

A role schema defines a common name for a set of roles, the role extent type, the logical position type and the position mapping functions, relating the real position with the logical position. Formally,

Definition 2 (Role schema): *A Role Schema is a tuple* $< r, ext, loc, m_{loc} >$ *where:*

- $r \in R$
- $ext \in REXT\_FT$
- $loc \in LPOS\_FT$
- $loc \subseteq_{ft} ext$
- $m_{loc} \in M$ *is a location mapping function for feature type loc.*

We denote with R_S *the set of role schemas and we assume that, given a role name* $r \in R$, *r is unique in* R_S.

An example of role schema is the tuple $<$ *CampusMember*, *Campus*, *Sector*, $m_{sector} >$ in which *CampusMember* is the name of the role; *Campus* is the type of role extent; *Sector* is the type of the logical position, represented by a sector of the campus; finally, m_{sector} is the position mapping function that maps a real position into one of the sectors of the campus. For the sake of readability, a role schema is written as *CampusMember* (*Campus*, *Sector*, m_{sector}).

Definition 3 (Role instance): *Given a role schema* $r_s \in R_S$, *an instance* r_i *of* r_s *is a pair* $< r, e >$ *where r is the name of the role in schema* r_s; *thus* $r = r_s.r$ *and* $e \in F$ *is a feature of type* r_s.ext. *The schema of* r_i *is denoted by SchemaOf* (r_i). *We denote with* R_I *the set of role instances for all role schemas.*

A role instance, e.g. $<$ *CampusMember*, *UniMi* $>$ is written as *CampusMember* (*UniMi*).

4.2.1 Permission

A permission is associated with each service. In our model, permissions can be associated either with the role schema and inherited by all role instances of the schema or directly with the role instances. Such different granularities are formalised by introducing two functions: *S_PrmsAssignment,* relating role schemas and permission sets; *I_PrmsAssignment* relating spatial roles, thus role instances, to specific permissions. Function *I_PrmsAssignment** is then introduced to combine the permissions directly assigned to spatial roles with the permissions inherited from their role schema.

Definition 4 (Permissions) *The set of permissions PRMS is defined as* $PRMS = 2^{(OPS \times OBJ)}$. *We also define:*

- SPA_S: $R_S \times PRMS$, *a many-to-many mapping permission-to-spatial role schema assignment relation.*
- *S_PrmsAssignment*: $R_S \rightarrow 2^{PRMS}$, *the mapping of spatial role schema onto a set of permissions. Given a role schema* r_s, *S_PrmsAssignment*$(r_s) = \{p \in PRMS \mid < r_s, p > \in SPA_S\}$.

- *SPA_I: $R_I \times PRMS$, a many-to-many mapping permission-to-spatial role instance assignment relation.*
- *$I\_PrmsAssignment$: $R_I \rightarrow 2^{PRMS}$ the mapping of spatial role instance onto a set of permissions. Given a role instance r_i, $I\_PrmsAssignment(r_i) = \{p \in PRMS \mid < r_i, p >\in SPA_I\}$.*
- *$I\_PrmsAssignment^*$: $R_I \rightarrow 2^{PRMS}$ such that given a role instance r_i, $I\_PrmsAssignment^*(r_i) = I\_PrmsAssignment(r_i)$□$S\_PrmsAssignment(SchemaOf\ (r_i))\}$. Hence the permissions of a role are those assigned to its schema plus those directly assigned to the instance.*

4.2.2 Users and session

Spatial roles are assigned to the users. The definition of the model for this part is conceptually analogous to that in RBAC. In particular, given a set of users, the following relations are defined: the many-to-many relation *SU A* relates users and role instances; the function *SR_AssignedUser* maps a role instance onto the set of users which can activate that role. More formally:

Definition 5 (Users) *We define:*

- *$SU\ A \subseteq U \times R_I$, a mapping user-to-spatial role instance assignment relation.*
- *$SR\_AssignedUser$: $R_I \rightarrow 2^U$, the mapping of spatial role instance onto a set of users. Formally $SR\_AssignedUser(< r, e >\in R_I) = \{u \in U | (u,< r, e >) \in SU\ A\}$.*

When a user logs in, a new session is activated and a number of roles are selected to be included in the session role set. Given a session *s*, the following two functions are defined: *SessionUser*(*s*) corresponds to the user of the session; *SessionRoles*(*s*) corresponds to the role that can be potentially activated in *s*. Formally:

Definition 6 (Sessions): *We define:*

- *$SessionUser$: $SES \rightarrow U$, the mapping from a session s to the user of s.*
- *$SessionRoles$: $SES \rightarrow 2^{R_I}$ with $SessionRoles\ (s) \subseteq \{< r, e >\in R_I | (SessionUser\ (s), < r, e >) \in SU\ A\}$.*

4.3 Access control mechanism

The session roles are the roles that the user of the session has selected. However, for a session role to be *enabled*, the user should logically be located within the space of the corresponding role extent. Therefore, depending on the user position during that session, only a subset of the session roles is enabled and permissions granted. In order to compute the logical position of a user playing a role *r* in a session, the location mapping function defined in the schema of *r* is applied to the user real position, provided by the external environment. Hence, if the logical position of the user is spatially contained in the extent of *r*, the role is *enabled* and thus the set of permissions assigned to the corresponding role is determined. Given a user's request, the access control mechanism determines whether the permission requested by the user belongs to the set of permission associated with the set of enabled roles *ER*. If it is the case the permission is granted, otherwise it is rejected.

The set of enabled roles *ER* in session *s* and real position *rp* is computed by the function *EnabledSessionRole* (*s*, *rp*).

Definition 7 (Authorisation control function): *An access request is a tuple ar* = ⟨*s*, *rp*, *p*, *o*⟩ ∈ *SES* × *RPOS* × *OPS* × *OBJ*. *ar can be satisfied at position rp* ∈ *RPOS if*

$$(p,o) \in \bigcup_{y \in EnabledSessionRoles(s,rp)} I\_PrmsAssignment^*(y).$$

4.4 Hierarchical GEO-RBAC

The Hierarchical GEO-RBAC introduces a number of additional concepts, in particular the notions of: (a) schema and instance hierarchy; (b) the notion of *replaceable* and *non-replaceable role*. These aspects are formally defined as follows.

4.4.1 Schema and instance hierarchy

A schema hierarchy is a partial order defined over the set of role schemas. The function *S_AuthorisedPrms* is defined, which computed the set of permissions granted to and inherited by a role.

Definition 8 (Schema hierarchy): *Let R_S be the set of role schemas. A schema hierarchy RHs is defined as follows:*

1 *$RH_s \subseteq R_S \times R_S$ is a partial order over R_S, denoted by $\preceq_S$. If $r_{s_1} \preceq_S r_{s_2}$ holds, then the role extent type in r_{s_2} is contained in the role extent type of r_{s_1} and similarly the logical position type in r_{s_2} is contained in the role extent type of r_{s_1}, hence $r_{s_2}.ext \subseteq_{ft} r_{s_1}.ext$, and $r_{s_2}.loc \subseteq_{ft} r_{s_1}.loc$.*

2 *$S\_AuthorisedPrms$: $R_S \rightarrow 2^{PRMS}$ such that, given a role schema r_s, $S\_AuthorisedPrms(r_s)$ returns all permissions assigned to r_s and to all its ancestors, that is $S\_AuthorisedPrms\,(r_s) = \{p \in PRMS \mid r'_s \preceq_S r_s, < r'_s, p > \in SPA_S\}$.*

An instance hierarchy is a partial order over the set of role instances. The function *I_AuthorisedPrms* computes the set of permissions granted to a role instance. The function *I_AuthorisedUsers* returns the set of user assigned to a role instance. Note that, however, a relationship is defined between the two hierarchies. The definition of instance hierarchy is given as follows.

Definition 9 (Instance hierarchy): *Let R_I be the set of role instances. An instance hierarchy RH_i is defined as follows:*

1 *$RH_i \subseteq R_I \times R_I$, a partial order over R_I, denoted by $\preceq_i$. The following properties hold:*

- *$<r_1, e_1> \preceq_i <r_2, e_2>$ holds if $SchemaOf(<r_1, e_1>) \preceq_s SchemaOf(<r_2, e_2>)$ and $LocObj(e_2) \subseteq LocObj(e_1)$.*

- *Given* $<r_2, e_2>$ *and* $RS_2 = SchemaOf(<r_2, e2>)$ *if* $RS_1 \preceq RS_2$*, then a role* $<r_1, e_1>$ *exists with* $RS_1 = SchemaOf(<r_1, e_1>)$.

2 *I_AuthorisedPrms*: $R_I \rightarrow 2^{PRMS}$ *such that, given a role instance* r_i*, I_AuthorisedPrms* (r_i) *returns all permissions assigned to* r_i *and to all its ancestors, that is* $I\_AuthorisedPrms\ (r_i) = \{p \in PRMS \mid r_i' \preceq_i r_i, p \in I\_PrmsAssignment^*\ (ri')\}$.

3 *I_AuthorisedUsers*: $R_I \rightarrow 2^U$ *such that, given a role instance* r_i*, I_AuthorisedUsers* (r_i) *returns all users assigned to* r_i *and to all its descendants, that is* $I\_AuthorisedUsers\ (r_i) = \{u \in U \mid r_i \preceq_i^* r_i', < u, r_i' > \in SU\ A\}$.

4.4.2 Representation of R-roles and NR-roles

The second aspect we consider is the formalisation of the replacement property. The *replacement property* is specified both in the role schema, to make it possible applying the property to all the instances of a role, and in the role instance definition, in order to characterise the single instance.

The role schema is thus extended with the *dist* attribute, which defines whether a role is a R-role or a NR-role. All the instances sharing the same schema inherit the same distance value. To enhance flexibility, the value of the property can, however, be specified for single instances also. The structure of the role instance is thus extended with an attribute labelled *dist* which, if not NULL, overrides the value of the corresponding attribute defined at schema level. The definition of the extended schema and role instance is given next.

Definition 10 (Extended role schema): *An extended role schema is the tuple* $< r, ext, loc, m_{loc}, dist >$ *where:* $dist \in N$ *is the distance which indicates whether the role is replaceable or not;* $dist = 0$ *means that the role is a NR-role,* $dist > 0$ *means that the role is a R-role which can be replaced by any ancestor at a maximum distance dist from the role.*

Consider the following example based on the role graph in Figure 1a. Assume a possible schema for role *E*: $< E, Ext_E, Loc_E, m_E, 1 >$. In this case, role *E* is a R-role since the distance is 1. It means that, unless differently specified, all role instances of *E* have the same value for the replacement property. Specifically, the instances of *E* can be replaced by the role instances at most at distance 1, that is those denoted as *B* and *C*. Note that a single role can be replaced by one or more roles. Further, a role cannot be replaced by the ⊥ role.

Definition 11 (Extended role instance): *Given a role schema* r_s*, an instance* r_i *of* r_s *is a triple* $< r, e, dist >$ *where: r is the name of the role in schema* r_s*;* $e \in F$ *is a spatial feature of the type specified in the role schema, that is* $r_s.ext$*;* $dist \in N$ *the distance, as defined earlier. It should be noticed that the distance property for the specific instance or undefined* (NULL).

4.5 Constrained GEO-RBAC

We now present the formal representation of the SoD constraints. We recall that a SoD constraint is defined as a pair (*role_set, n*). From Section 4.4, we recall also that in our model the following classes of constraints have been devised:

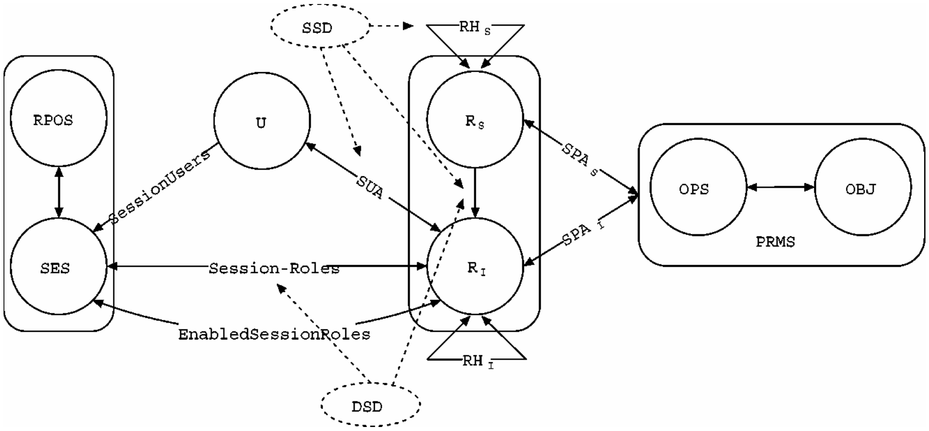
1 non-spatial constraint at instance level

2 non-spatial constraint at schema level

3 spatial constraint at schema level

The general structure of the model is illustrated in Figure 3. We extend the graphical representation adopted in RBAC by considering the constraints at schema level also.

Figure 3 Constrained GEO-RBAC



We first provide the definition of the constraints interpreted as static SoD constraints. Then we describe the extension to the dynamic SoD. Some system functions are introduced to support the subsequent definitions. In particular, we introduce the functions:

- *Ext*(*S*): returns the set of role instances of schema S.
- *AreRelated* (r_1, r_2, *rel*): returns true if the spatial relationship *rel* holds between the geometries of the extents of roles r_1 and r_2.

4.5.1 Static separation of duty – SSD

The first type of constraints we consider are the *non-spatial constraints defined at instance level*. These constraints are those which more closely resemble the classical RBAC SSD constraints.

Definition 12 (Non-spatial constraint at instance level): *Let RoleSet* = {r_1, ..., r_k} *be a set of role instances. The constraint* (*RoleSet*, *n*), $n > 1$, *is satisfied if for any subset h of RoleSet having cardinality at least n*; *no user has been assigned to all roles in the subset. Formally:*

$$\forall h \subseteq RoleSet, | h | \geq n \Rightarrow \bigcap_{r \in h} SR\_AssignedUsers(r) = \emptyset$$

Example 1 *Consider the set RoleSet* = {*CampusMember*(*A*),*CampusMember*(*B*), *CampusMember*(*C*)}. *According to the Definition 12, the constraint* (*RoleSet*, 2) *means* *that an individual cannot be a member of two or more campuses among those specified in RoleSet. Note that the roles can be instances of either the same or different schemas.*

The next type of constraints we consider are the *non-spatial constraints defined at schema level*. Basically, a constraint states that an individual cannot play roles from some given schemas. We distinguish two cases: in the first case, the set of role schemas which is specified in the constraint consists of two or more schemas; in the second case, the constraint is specified on a single schema.

Definition 13 (Non-spatial constraint at schema level): *Let SchemaSet* = $\{SR_1, \ldots, SR_m\}$ *be a set of role schemas with* $SR_i \neq SR_j$ *for* $i \neq j$. *We distinguish two cases depending on the cardinality of SchemaSet. The constraint* (*SchemaSet*, *n*), $n > 1$, *is satisfied if:*

Case A: $|\,SchemaSet\,| \geq 2$

for any subset of SchemaSet having cardinality at least n; no user has been assigned at least one role for each schema in the subset.

Case B: $|\,SchemaSet| = 1$

Let $\{SR\}$ *be the unique schema in SchemaSet. Then no user has been assigned at least n roles of SR.*

Formally:

Case A

$$\forall s \cong \{SR_i, \ldots, SR_j\} \subseteq SchemaSet, |\, s \geq n \Rightarrow \bigcap_{1 \leq k \leq j} \Big(\bigcup_{r \in Ext(SR_k)} SR\_AssignedUsers(r)\Big) = \emptyset$$

Case B

$$\forall h \subseteq Ext(SR), |\, h \,| \geq n, \Rightarrow \bigcap_{r \in h} SR\_AssignedUsers(r) = \emptyset$$

Example 2 *Consider the role schema* $\{CampusDirector\ (Campus, Sector, m_{Sector})\}$. *The constraint* (*CampusDirector*, 2) *means that an individual cannot be a director of two or more different campuses. Notice that since the schema denotes a set of roles, the definition is similar to Definition 12.*

The last type of constraints we discuss are the *spatial constraints defined at schema level*. In this case the constraint states that an individual cannot play any pair of roles of some given schemas satisfying a specified spatial condition. The spatial relationships we consider in particular are the topological relationships, because they are more relevant in real applications.

Definition 14 (Spatial constraint at schema level): *Let* SR_1 *and* SR_2 *be two role schemas and Rel a spatial relationship. The constraint* (SR_1, SR_2, *Rel*) *is satisfied if no user is assigned to any role of schema* SR_1 *and any role of schema* SR_2 *satisfying the spatial relationship Rel. Formally:*

$$\left.\begin{array}{l} \forall x \in Ext(SR_1), \forall y \in Ext(SR_2) \\ AreRelated(x, y, Rel) = True \end{array}\right\} \Rightarrow SR\_AssignedUser(x) \bigcap SR\_AssignedUsers(y) = \emptyset$$

Note that in case of role hierarchies, the function that computes the set of users which have been assigned a given role is the function *SR_AuthorisedUsers* (in place of *SR_AssignedUsers*).

Example 3 *The constraint:*

> (*CampusTeacher* (*Campus*, *Sector*, m_{Sector}), *CampusStudent* (*Campus*, *Sector*, m_{Sector}), *Overlap*)

means that an individual cannot be assigned to two roles, teacher and student, over overlapping regions.

4.5.2 Dynamic separation of duty

Now we define the meaning of the constraints interpreted as DSD constraints. Basically, what changes under this interpretation with respect to the static one is that the constraints concern the roles which are activated in a session. The meaning of each class of constraints is reported in the following:

Definition 15 (Non-spatial constraint at instance level): *Let RoleSet* = {r_1, ..., r_k} *be a set of role instances. The constraint* (*RoleSet*, *n*), $n > 1$, *is satisfied if there is no session in which at least n roles from the role set have been activated. Formally:*

$$\forall t \in SES, \forall h \subseteq RoleSet, h \subseteq SessionRoles(t)\} \Rightarrow | h | < n$$

Example 4 *With reference to the role set presented in the Example 1, the constraint* (*RoleSet*, 2) *means that an individual cannot activate two or more campus membership roles among those specified in RoleSet. Note that the user can be assigned different roles of the schema CampusMember.*

Definition 16 (Non-spatial constraint at schema level): *Let SchemaSet* = {SR_1, ..., SR_m} *be a set of role schemas with SRi ≠ SR_j for i ≠ j. We distinguish two cases depending on the cardinality of SchemaSet. The constraint* (*SchemaSet*, *n*), $n > 1$, *is satisfied if:*

Case A: $| SchemaSet | \geq 2$

There is no session in which at least n roles, one for each different schema from SchemaSet, have been activated.

Case B: $| SchemaSet | = 1$

Let {*SR*} *be the unique schema of SchemaSet. There is no session in which at least n different role instances of SR have been activated.*

Formally:

Case A:

$$\left.\begin{array}{r} \forall t \in SES, \forall s \cong \{SR_i, ..., SR_j\} \subseteq SchemaSet, \\ \forall h \cong \{r_i, ..., r_j\}, r_i \in Ext(SR_i), ..., r_j \in Ext(SR_j), \\ h \subseteq SessionRoles(t) \end{array}\right\} \Rightarrow | h | < n$$

Case B:

$$\forall t \in SES, \forall h \subseteq Ext(SR), h \subseteq SessionRoles(t) \Rightarrow | h | < n$$

Example 5 *With reference to the schema set presented in the Example 2, the constraint* (*SchemaSet*, 2) *means that an individual cannot activate two or more* CampusDirector *roles.*

Definition 17 (Spatial constraint at schema level): *Let SR_1 and SR_2 be two role schemas and Rel a spatial relationship. The constraint* (SR_1, SR_2, *Rel*) *is satisfied if there is no session in which roles x and y of schema, SR_1 and SR_2, respectively, satisfying the spatial relationship Rel have been activated simultaneously. Formally:*

$$\left.\begin{array}{l}\forall x \in Ext(SR_1), \forall y \in Ext(SR_2) \\ \forall t \in SES, AreRelated(x, y, Rel) = True\end{array}\right\} \Rightarrow \{x, y\} \not\subset SessionRoles(t)$$

Example 6 *With reference to the schema set presented in Example 3, the constraint means that an individual can be assigned to the roles of teacher and student even if they are over overlapping regions, but he/she cannot activate more than one of both these roles.*

Finally, we can observe that the class of spatial constraints at schema level, defined both at static and dynamic level, significantly increase the expressivity of the model because the SoD constraints are specified over sets of roles that are defined intensionally based on spatial conditions on role extents.

5 Open issues: location privacy

Data security in Enterprise LBS is an important issue and *GEO-RBAC* represents a possible approach. However, with the large deployment of LBS and the improvements in accuracy and precision of positioning systems, another important issue is related to *location privacy*. *Location privacy* can be defined as the ability to prevent other parties from learning one's current or past location (Beresford and Stajano, 2003). A threat to location privacy thus occurs when an adversary can obtain an individual's location information *and* can identify the individual.

We illustrate the concept through an example. Consider a table describing (a) the identifier of users and (b) the position of users, represented at the level of precision enabled by the positioning technology, for instance, as points of coordinates (*x,y*). Even though the identity of the user is striped off, the information about *who is where* can still be determined if location can spatially be associated with an external database. For example, if we know that somebody is at point *P* and from another database that in the region *A* containing *P* there is a unique individual, say *S*, we can infer that *S* is located in *P*. In this case, data association is simply computed through an operation of spatial join.

To reduce the risk of threats to location privacy, a possible approach is based on the use of *k-anonymity* techniques (Sweeney, 2002). In conventional databases, *k-anonymisation* is a property defined for tables. In particular, given a table *t* containing the user-identifying properties and sensitive data (say, medical diseases), *t* is defined to be *k-anonymous* when each value of the identifying properties appears at least *k* times in *t*. For example, if the identifying property is the postal code, each value of the postal code can be perturbed by reducing the number of digits so that there are at least *k* tuples with the same value of postal code. In this way, the actual identity of user is hidden in a set of peers. Further, parameter *k* defines a metric for the degree of privacy.

Recently, extensions of such techniques have been proposed to deal with *location privacy*. In particular, in Gruteser and Grunwald (2003) the notion of k-anonymity has been reformulated as follows: *a subject is k-anonymous with respect to location information, if and only if the location information is indistinguishable from the location information of at least k-*1 *other subjects*. The method which has thus been proposed to make locations undistinguishable is to replace the actual position with a coarse position. The coarse position is obtained by subdividing the space in rectangles each containing at least *k* elements. A similar approach, based on the idea of coarsening the position of the user but relying on a different technique, is proposed in Gedik and Liu (2005). In this case, the idea is to account not only for the spatial dimension but also for the temporal dimension. The coarsening operation is thus applied to the position in time and in place.

Research on location privacy is, though at early stage, rapidly progressing. One of the future challenges is the development of a unique framework for data security and location privacy support. In that perspective, one of the concepts that can probably be exploited to deal with both issues is that of spatial and temporal granularity.

We conclude with a final remark. So far, we have discussed location privacy focusing on the technical issues. However, the socio-technical aspects are a fundamental dimension of the problem (Thomas and Sandhu, 2004). In this perspective it would be interesting to consider if Enterprise LBS, because of the organisational setting, pose specific requirements with respect to location privacy. To our knowledge this issue has not been considered yet.

6 Related work

The development of spatially aware access control systems is an emerging research issue spanning several fields. In geographical information systems research area, the demand for spatially aware access control systems is primarily motivated by the increased concern for geographical information sharing. To our knowledge, the first access control model for geographical data has been proposed in Atluri and Mazzoleni (2002); Chun and Atluri (2000) only deals with satellite image maps. On the other hand, an access control system for geometric and vector-based spatial data has been proposed in Bertino, Damiani and Momini (2003). The model introduces the concept of spatial authorisation as an authorisation that can be defined only on portions of space. When an access request is made for an object, the system checks whether the requested object lies in the authorisation space and if this is the case, it grants the access. This model has been applied to support controlled access to spatial data on web. The underlying spatial data model is, however, relatively simple and does not address important issues such as the multi-granularity of spatial data. A similar architecture, but focused on XML-based representation of spatial data, has been proposed in Purevjii et al. (2004). A more complex spatial data model has been assumed in Belussi et al. (2004). In this work, an access control system is presented that allows the specification of authorisation rules to access complex-structured spatial data stored in a DBMS and organised according to multiple spatial representation levels and at multiple granularities. The system, however, does not deal with geographically bounded roles neither with mobile users. The position of users is considered in access control models securing mobile and context-aware applications. In IEEE 2003 Conference Hansen and Oleshchuk (2003 a,b) an extension of RBAC is proposed based on the notion of spatial role, intended as a role that is automatically activated when the user is in a given position. The space model is, however, very simple and targeted to wireless network applications. It consists of a set of adjacent cells and the position of the user is the cell or the aggregate of cells containing it. The spatial granularity of the position is thus fixed while the space is rigidly structured and the position itself does not have any semantic meaning but simply a geometric value. By contrast, in our model the granularity of the user position may depend on the role of the user; thus no assumption is made on the space layout. Moreover, the spatial dimension integrates geometric and semantic knowledge about the world. User position can be considered as a state variable in access control systems based on the notion of context (Covington, Moyer and Ahamad, 2000; Covington et al., 2001; Sampemane, Naldurg and Campbell, 2002; Cuppens and Miège, 2003). Of particular interest is the access control system proposed in Covington et al. (2001), Covington, Moyer and Ahamad (2000), introducing the concept of environment roles. Roles can be activated based on the value of conditions in the environment where the request has been made. Environmental conditions include time, location and other contextual information that is relevant to access control. If compared with GEO-RBAC, the concepts of role extent and user position are close to that of context variables. However, the mechanism of contexts is very general and does not account for the specificity of spatial information, such as the multi-granularity of position and the spatial relationships that may exist between the spatial elements in space. Moreover, in GEO-RBAC a common spatial data model is adopted in order to provide a uniform and standard-based representation of locational aspects that, notably, involve not only the roles but also the protected objects.

7 Conclusions

In this paper we have presented *GEO-RBAC*, an extension of the RBAC model addressing access control requirement in spatial and location-based information. Unlike other proposals of spatially aware access control models, GEO-RBAC relies on geo-spatial standards to model objects, user positions and geographically bounded roles, making the approach quite standard and flexible. Another important characteristic of the model is the ability to represent the concept of position at different levels of abstraction which are also role-dependent. This may be very useful to protect location privacy, because the position of the user can be specified at different levels of granularity and thus of positional uncertainty. Further, by introducing the concept of role schema, role extents and logical positions can be customised, depending on the function the role represents. Moreover, the user's roles are given a *status* which accounts for the mobility of the user when connected to the system. In addition to the Core GEO-RBAC, the model consists of two additional components: the Hierarchical GEO-RBAC and Constrained GEO-RBAC. The first introduces hierarchies that allow one role to inherit permissions from its ancestor roles, users from its descendant roles and roles to be enabled when descendant roles are. The second component adds to the model spatially aware SoD constraints to prevent conflicts of interest among the roles assigned to the user or activated during the user's session, based also on spatial criteria. As a result, GEO-RBAC constitutes a comprehensive model based on the spatial concepts that can be of effective support when deployed to regulate the access to location-aware services in mobile organisations. The set-theoretic specification of the model can then be turned into an operational specification. In Appendix, we describe a possible methodology for turning the GEO-

RBAC specification into a XML-based specification. As part of future work, we plan to investigate several directions. First, we plan to develop a distributed architecture for the access control functions and to provide support for k-anonymity. We also plan to integrate this model with the X-GTRBAC system (Bhatti et al., 2005), an XML-based temporal access control model based on RBAC, in order to obtain an access control system supporting the specification and enforcement of a rich set of context-based access control policies. Finally, we plan to develop encryption-based access control techniques specifically tailored to space-based access control policies.

Acknowledgements

The work of M.L. Damiani has been partially supported by the European project GeoPKDD 0“Geographic Privacy-aware Knowledge Discovery and Delivery” and by the italian MIUR “Estrazione e presentazione di conoscenza geografica con tecniche di data mining che salvaguardano la privacy”. This material is based upon work supported by National Science Foundation under Grant No. 0430274 and the sponsors of CERIAS.

Appendix

In this appendix we report the XML-based specification of Core GEO-RBAC named XGEO-RBAC. The motivation for XGEO-RBAC is that it provides a running specification of the model, and at the same time represents a component that can be integrated into web-based architectures of innovative spatial applications. Here we focus on the methodology that has been developed for turning Core GEO-RBAC into XML.

The proposed approach is based on two key design choices: first, specification has been developed not directly in XML but in GML (Geographic Mark-Up Language), the language built on XML which encodes spatial data organised in terms of OGC standards. Second, in order to enable a more systematic approach and also to provide an intuitive graphical notation of the operational model, that otherwise if expressed directly in GML might be difficult to appreciate and customise, we have used UML (Unified Modelling Language) (UML, 2005), specifically the UML profile in ISO/TC211 (2001), for the object-based representation of GEO-RBAC. In this way, first the theoretical model is mapped onto a UML class diagram and then the class diagram is encoded in GML. In this final section, we first briefly present XML, GML and UML. Next we introduce the UML profile for class diagram. The mapping rules from our UML profile to GML that are compliant with current standards, are then briefly presented.

The notations: XML, GML and UML

XML. The Extensible Mark-up Language (W3C, 2004) has become a standard not only for data communication over internet but also for the development of interoperable data models. To achieve this, XML provides a metalanguage that allows the definition of mark-up tags to define custom documents and technologies for their interpretation. A XML document has a logical structure that notably includes *elements*. Elements are delimited by start-tags and end-tags. Further, elements may have a structure consisting of nested elements. Such a structure may be constrained by the *type* of the element. An *XML Schema* defines the common logical structure of a set of XML documents. It consists of components such as type definitions and element declarations that are used to validate XML documents. An 'instance document' is the XML document that conforms to a particular schema.

GML. Geography Mark-up Language (ISO/TC211, 2004) is an XML grammar written as XML schema for the modelling, transport and storage of geographic information. GML defines the various entities of interest such as features, geometries and spatial reference systems through a hierarchy of *GML objects* with identity and properties. GML objects correspond to XML elements of a type derived directly or indirectly from a general type *AbstractGMLType*. *Properties* of GML objects are XML children elements of a GML object. A *GML Application Schema* is an XML schema written according to the GML rules and defines a vocabulary of geographic objects for a particular domain of discourse.

UML. The Unified Modelling Language is a widely used notation for object-oriented specifications (UML, 2005). It is applied in a variety of fields, ranging from software engineering to databases and business modelling. A UML specification consists of one or more diagrams describing both static and dynamic aspect along the various phases of the system life cycle, from requirement analysis to implementation. In the geographical domain, UML has been adopted as reference notation for the development of conceptual schemas (or *Application Schemas*) of geographical information. For that purpose, a UML profile (*standard UML profile* for short) has been defined allowing the specification of spatial entities in terms of class diagrams. Recently, rules for mapping the UML application schema onto a GML application schema have been explicitly defined (ISO/TC211, 2004).

The UML specification

The first step of the process leading to the XML encoding of the core component is to represent the elements of the model in terms of UML classes. We use in particular the following constructs:

- *Classes.* We consider classes that represent *object types* and *feature types*. Object and feature type classes can be also *abstract*; therefore, instances cannot be created for them (abstract classed are in italics). A specific class of objects is the class of geometric objects (class GM_object).
- *Relationships*. The relationships we consider are *generalistion, ordinary association and specialisation by restriction*. Generalisation (→) and ordinary associations (→) derive from standard UML. A role name is thus required on every navigable association. We assume that '1' is the default cardinality for a navigable association. *Specialisation by restriction* means that if a class, say *C*, specialises a parent class, say *P*, then *C* can have attributes whose values are a subset of the values of the same attribute in *P*. For example, if the parent class has a property *Shape* of type *Geometry*, then a specialised class can have the same property but of type *Surface*. Since, as we will see, the construct is useful during the GML encoding, we include it

explicitly in the sub-profile, although it is not a native construct of UML and would require an explicit constraint. Graphically, it is denoted by a generalisation arrow marked with *R*.

The application schema of *Core GEO-RBAC* is shown in Figure 4. The schema represents in terms of classes and relationships the concepts of the Core GEO_RBAC, thus the concepts of *spatial role*, *spatial role schema*, *user*, *session*, *permission* and *user position*. More specifically:

- *Spatial role and spatial role schema*. The abstract class *SpatialRole* defines the general properties of a role. In particular, the extent of the role as well as the permissions assigned to the role are specified in terms of associations with the class of role extents and the permission class, respectively. The location-mapping function *m_loc,* which maps the real position onto the logical position, is specified as an operation of the *SpatialRole* class. Further, in order to account for the other (meta) properties of roles and in particular the permissions that are granted to all the instances of the schema (*s-permissions*), we introduce the fictitious class *RoleSchema* that specifies, through an association (*s_has_perm*), the set of *s-permissions*.

- *User and sessions*. The *User* class defines, through an association with the *Spatial Role* class (*u_has_role*), the set of roles which are assigned to the user. The *Session* class describes the users of the session. Further, since the real position is assumed to be dynamically acquired through an external device or program, the *Session* class contains, among the operations, the function (*m_rpos*) to fetch the real position of the session user. *Session* is linked to the class *SessionRole* which specifies the set of roles that are active in the session. The attribute *is_enabled* in *SessionRole* describes the status of each active role (i.e. *enabled/disabled*).

- *Permissions*. Permissions are modelled through the *Permission* class. Such a class is related through two associations to the *Operation* and the *Resource* class, respectively. The *Resource* class models the domain objects that need to be protected. The *Operation* class models the operations that the user can require over the resources.

- *Real and logical position*. The real position, as it consists of a geometric object, is defined as a specialisation of the abstract class *GM_Object*. An instance of real position can be, for example, a point or a polygon. The notion of logical position is introduced through the *LogicalPosition* abstract feature type class.

Once defined the UML application schema, the next step is to map it onto a GML application schema in accordance with the rules introduced by the current standard (ISO/TC211, 2004). The mapping, however, concerns only the structural properties of classes. To exemplify, in Figures 5 and 6 we show the mapping of the application role *TaxiDriver* defined over the extent *UrbanZone*. In the former figure, two GML features, *city* and *UrbanZone*, are specified as role extents; in the latter figure the role *TaxiDriver* is specified as spatial role with extent *UrbanZone*.

Figure 4 Core GEO-RBAC UML application schema

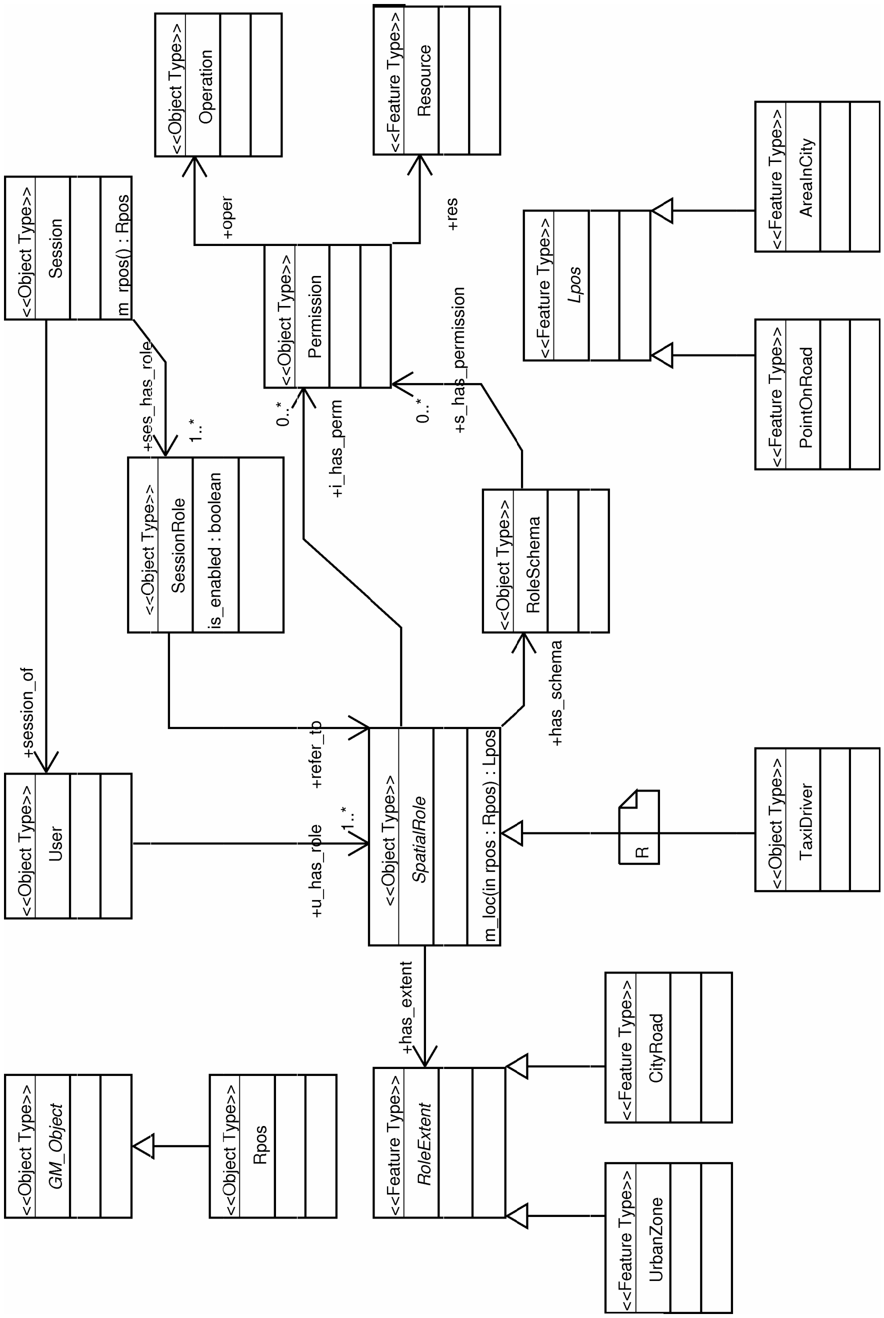
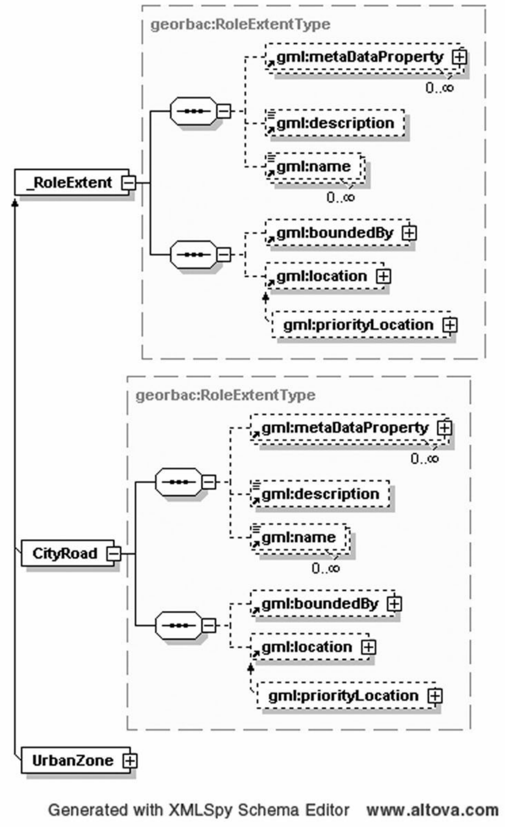
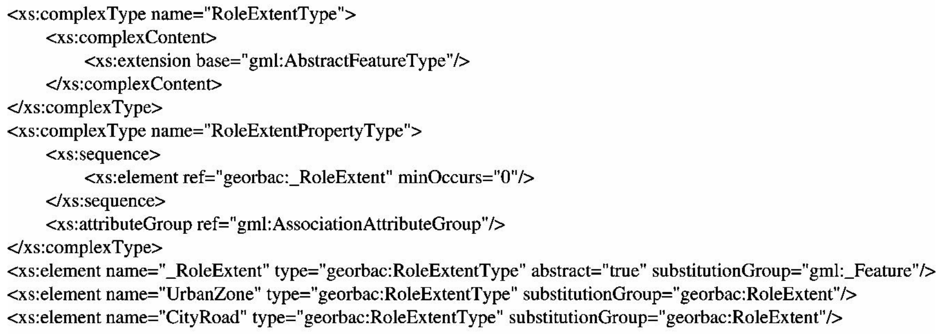


Figure 5 Role extent

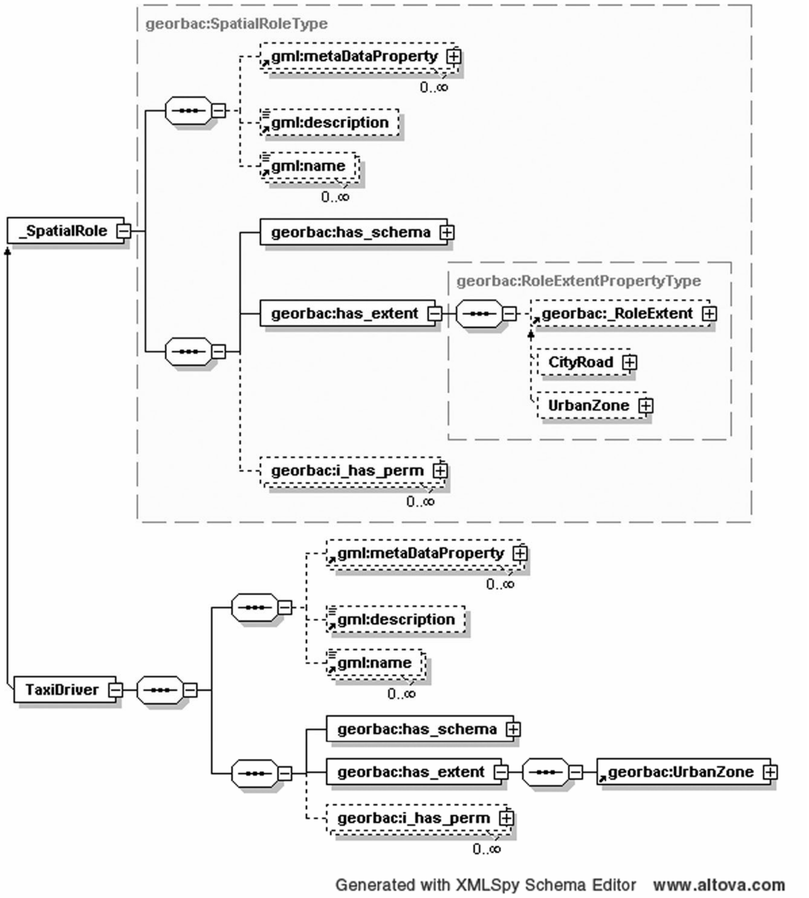


(a) Abstract role extent element diagram

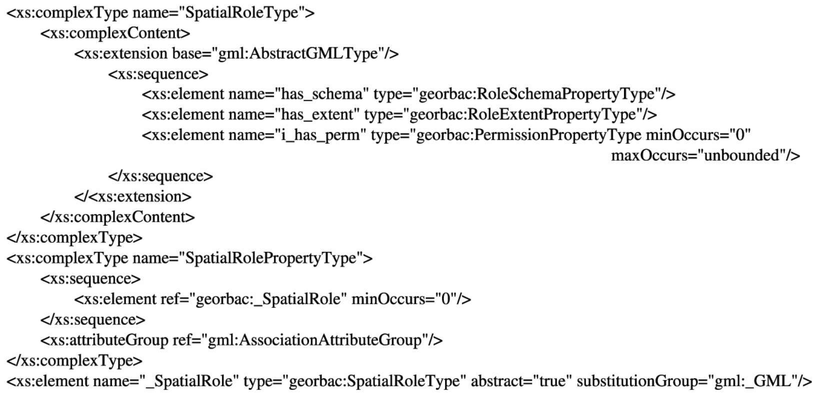

```
<xs:complexType name="RoleExtentType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType"/>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="RoleExtentPropertyType">
    <xs:sequence>
        <xs:element ref="georbac:_RoleExtent" minOccurs="0"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<xs:element name="_RoleExtent" type="georbac:RoleExtentType" abstract="true" substitutionGroup="gml:_Feature"/>
<xs:element name="UrbanZone" type="georbac:RoleExtentType" substitutionGroup="georbac:RoleExtent"/>
<xs:element name="CityRoad" type="georbac:RoleExtentType" substitutionGroup="georbac:RoleExtent"/>
```

(b) GML code

Figure 6 Spatial role



(a) Abstract spatial role element diagram



```
<xs:complexType name="SpatialRoleType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractGMLType"/>
            <xs:sequence>
                <xs:element name="has_schema" type="georbac:RoleSchemaPropertyType"/>
                <xs:element name="has_extent" type="georbac:RoleExtentPropertyType"/>
                <xs:element name="i_has_perm" type="georbac:PermissionPropertyType minOccurs="0"
                                                                        maxOccurs="unbounded"/>
            </xs:sequence>
        </<xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="SpatialRolePropertyType">
    <xs:sequence>
        <xs:element ref="georbac:_SpatialRole" minOccurs="0"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<xs:element name="_SpatialRole" type="georbac:SpatialRoleType" abstract="true" substitutionGroup="gml:_GML"/>
```

(b) GML code

Following ISO/TC211 (2004), with some variations, the mapping rules are:

1 *Classes*. The classes with stereotype ≪*ObjectType*≫ are mapped onto global elements of the type derived from the general type *gml:AbstractGMLType*. Similarly, the classes with stereotype ≪*FeatureType*≫ are mapped onto elements of the type derived from the general type *gml:AbstractFeatureType*.

2 *Generalisation*. Generalisation is realised by using the mechanism of derivation by extension.

3 *Specialisation by restriction*. Specialisation by restriction is realised by deriving-by-restriction a new type for the more specific class. Figure 6 shows the *TaxiDriver* element of type derived-by-restriction from that of *SpatialRole*. In this case, the restriction concerns the range of values of the role extent property.

4 *Associations*. Association between classes is realised following the patterns in ISO/TC211 (2004).

References

Atluri, V. and Mazzoleni, P. (2002) 'A uniform indexing scheme for geo-spatial data and authorizations', Paper presented at the *16th Conference on Data and Application Security (IFIP TC11/WG11.3)*, Cambridge, pp.207–218. In proceedings.

Belussi, A., Bertino, E., Catania, B., Damiani, M.L. and Nucita, A. (2004) 'An authorization model for geographical maps', Paper presented at the *12th International Symposium of ACM GIS*, Washington, DC, pp.82–91. In proceedings.

Beresford, A. and Stajano, F. (2003) 'Location privacy in pervasive computing', *IEEE Pervasive Computing*, Vol. 3, pp.46–55.

Bertino, E., Bonatti, P. and Ferrari, E. (2001) 'TRBAC: a temporal role-based access control model', *ACM Transactions on Information and System Security (TISSEC)*, Vol. 4, pp.191–233.

Bertino, E., Damiani, M.L. and Momini, D. (2003) 'An access control system for a web map management service', Paper presented at the *14th International Workshop on Research Issues in Data Engineering (RIDE-WS-ECEG)*, Boston, MA, pp.33–39. In proceedings.

Bertino, E., Catania, B., Damiani, M.L. and Perlasca, P. (2005) 'GEO-RBAC: a spatially aware RBAC', Paper presented at the *International Symposium of ACM Access Control Models and Technologies (SACMAT 2005)*, Stockholm, pp.29–37. In proceedings.

Bhatti, R., Ghafoor, A., Bertino, E. and Joshi, J. (2005) 'X-GTRBAC: an XML-based policy specification framework and architecture for enterprise-wide access control', *ACM Transactions on Information and System Security (TISSEC)*, Vol. 8, pp.187–227.

Chun, S.A. and Atluri, V. (2000) 'Protecting privacy from continuous high-resolution satellite surveillance', Paper presented at the *14th IFIP 11.3 Working Conference on Database Security*, Schoorl, pp.233–244. In proceedings.

Clementini, E., Di Felice, P. and van Oosterom, P. (1993) 'A small set of formal topological relationships suitable for end-user interaction', Paper presented at the *LNCS 692: Proceedings of the 3rd International Symposium on Advances in Spatial Databases (SSD'93)*, Singapore, pp.277–295. In proceedings.

Covington, M., Long, W., Srinivasan, S., Dev, A.K., Ahamad, M. and Abowd, G.D. (2001) 'Securing context-aware applications using environment roles', Paper presented at the *6th ACM Symposium on Access Control Models and Technologies*, Chantilly, pp.10–20. In proceedings.

Covington, M., Moyer, M. and Ahamad, M. (2000) 'Generalized role-based access control for securing future applications', Paper presented at the *23rd National Information Systems Security Conference (NISSC 2000)*, Baltimore. In proceedings.

Cuppens, F. and Miège, A. (2003) 'Modelling contexts in the Or-bac model', Paper presented at the *19th Annual Computer Security Applications Conference (ACSAC'03)*, Las Vegas, pp.416–427. In proceedings.

Ferraiolo, D., Sandhu, R., Gavrila, S. and Kuhn, R. and Chandramouli, R. (2001) 'Proposed NIST standard for role-based access control', *ACM Transactions on Information and System Security (TISSEC)*, Vol. 4, pp.224–274.

Gedik, B. and Liu, L. (2005) 'Location privacy in mobile systems: a personalized anonymization model', Paper presented at the *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, Columbus, pp.620–629. In proceedings.

Gruteser, M. and Grunwald, D. (2003) 'Anonymous usage of location-based services through spatial and temporal cloaking', Paper presented at the *First ACM/USENIX International Conference on Mobile Systems, Applications and Services (Mobisys 2003)*, San Francisco. In proceedings.

Hansen, F. and Oleshchuk, V. (2003a) 'Spatial role-based access control model for wireless networks', Paper presented at the *IEEE Vehicular Technology Conference (VTC2003-Fall)*, Orlando. In proceedings.

Hansen, F. and Oleshchuk, V. (2003b) 'Srbac: a spatial role-based access control model for mobile systems', Paper presented at the *8th Nordic Workshop on Secure IT Systems (Nordsec 2003)*, Gjøvik, pp.129–141. In proceedings.

ISO/TC211 (2001) 'Draft Technical Specification 19103, geographic information – conceptual schema language'.

ISO/TC211 (2003) 'Technical Specification 19107, geographic information – spatial schema'.

ISO/TC211 (2004) 'Technical Specification 19136, geographic information – geography Markup Language'.

Moffett, J.D. and Lupu, E.C. (1999) 'The uses of role hierarchies in access control', Paper presented at the *4th ACM workshop on Role-based Access Control*, Fairfax, pp.153–160. In proceedings.

Nyanchama, M. and Osbornm, S. (1999) 'The role graph model and conflict of interest', *ACM Transactions on Information Systems Security (TISSEC)*, Vol. 2, pp.3–33.

Purevjii, B., Amagasa, T., Imai, S. and Kanamori, Y. (2004) 'An access control model for geographic data in an XML-based framework', Paper presented at the *2nd International Workshop on Information Systems Security (WOSIS 2004)*, Porto, pp.251–260. In proceedings.

Samarati, G. and Sweeney, L. (1998) 'Generalizing data to provide anonymity when disclosing information', Paper presented at the *17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of database systems*, Seattle, p.188. In proceedings.

Sampemane, G., Naldurg, P. and Campbell, R.H. (2002) 'Access control for active spaces', Paper presented at the *18th IEEE Annual Computer Security Applications Conference (ACSAC'02)*, Las Vegas, pp.343–352. In proceedings.

Sandhu, R., Ferraiolo, D. and Kuhn, R. (2000) 'The NIST model for role-based access control: towards a unified standard', Paper presented at the *5th ACM Workshop on Role-Based Access Control*, Berlin, pp.47–63. In proceedings.

Shekhar, S., Vatsavai, R., Ma, X. and Yoo, J. (2004) 'Navigation systems: a spatial database perspective', In J. Schiller and A. Voisard (Eds), *Location-Based Services*. Morgan Kaufmann Publishers, ISBN: 1–55860–929–6.

Spiekermann, S. (2004) 'General aspects of location-based services', In J. Schiller, and A. Voisard, (Eds.), *Location-Based Services*. Morgan Kaufmann Publishers, 2004.

Sweeney, L. (2002) 'Achieving k-anonymity privacy protection using generalization and suppression' *Int. J. on Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 10, pp.571–588.

Thomas, R. and Sandhu, R. (2004) 'Models, protocols and architectures for secure pervasive computing: challenges and research directions', Paper presented at the *2nd Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW'04)*, Orlando, pp.164–170. In proceedings.

UML (2005) 'Unified modeling language', Available at: *http://www.uml.org*

W3C (2004) 'Extensible markup language (XML) 1.1 – W3C Recommendation', Available at: *http://www.w3.org/TR/2004/REC-xml11-20040204*