GEO-RBAC: A Spatially Aware RBAC

Elisa Bertino
Department of Computer Science
Purdue University
West Lafayette,IN 47907, US
bertino@cerias.purdue.edu

Barbara Catania
Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova
Via Dodecaneso 35, Genova, Italy
catania@disi.unige.it

Maria Luisa Damiani, Paolo Perlasca
Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39/41, Milano, Italy
{damiani,perlasca}@dico.unimi.it

ABSTRACT

Securing access to data in location-based services and mobile applications requires the definition of spatially aware access control systems. Even if some approaches have already been proposed either in the context of geographic database systems or context-aware applications, a comprehensive framework, general and flexible enough to cope with spatial aspects in real mobile applications, is still missing. In this paper, we make one step towards this direction and we present GEO-RBAC, an extension of the RBAC model to deal with spatial and location-based information. In GEO-RBAC, spatial entities are used to model objects, user positions, and geographically bounded roles. Roles are activated based on the position of the user. Besides a physical position, obtained from a given mobile terminal or a cellular phone, users are also assigned a logical and device independent position, representing the feature (the road, the town, the region) in which they are located. To make the model more flexible and re-usable, we also introduce the concept of role schema, specifying the name of the role as well as the type of the role spatial boundary and the granularity of the logical position. We then extend GEO-RBAC to cope with hierarchies, modeling permission, user, and activation inheritance.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial Databases and GIS*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Management, Security, Theory

Keywords

GIS, Authorization model, Access control, Location-based services

1. INTRODUCTION

The widespread deployment of location-based services and mobile applications as well as the increased concern for the management and sharing of geographical information in strategic applications like environmental protection and homeland security have resulted in a strong demand for spatially aware access control systems. These application domains pose interesting requirements against access control systems. In particular, the permissions assigned to users depend on their position in a reference space; users often belong to well defined categories; objects to which permissions must be granted are located in that space; access control policies must grant permissions based on object locations and user positions.

As an example, consider an information service providing real time traffic information to car drivers. Suppose that the set of users includes different categories of drivers, such as tourists, taxi drivers and police. Each category of drivers needs to access different information resources. For example taxi drivers should be allowed to get detailed traffic information about roads as well as notification about the position of possible traffic jams or accidents occurred along roads. Both traffic and jam information refer to well defined locations in the reference space. In this context, we may need to specify for example that a taxi driver could be allowed to notify accidents only when driving along some major roads within the city and only about accidents that have occurred within given distance from the his/her position.

To deal with the requirements listed above, an access control model with spatial capabilities is needed. Since in location-aware applications users are often grouped in distinct categories (tourists, taxi drivers and police in the previous example) role-based access control models (RBAC models) [9, 17] represent a reasonable choice. Various role-based and spatially aware access control systems have been proposed for securing access to spatial data stored in a spatial DBMS or for securing access to location-aware applications [6, 7, 8, 10, 11, 16]. Even though some preliminary proposals have been reported adding contextual information, such as spatial and temporal information to access control mechanisms, such approaches are simplistic and do not account for several of the requirements we have devised such as multigranularity of position and relationships in space.

In this paper, we overcome those limitations by proposing a comprehensive spatial framework for an access control system securing access to spatial data in location-aware applications. Such a model, called GEO-RBAC, is an extension of the RBAC model with the concept of spatial role and supports the homogeneous representation of all spatial aspects involving roles, objects and contextual information such as user position. The spatial model we adopt is compliant with OGC (Open GeoSpatial Consortium) [12]. Thus, it is based on the notion of feature type (a road, a town, a region) and feature, as instance of a given feature type (road A10, Milan, Lombardy). Features have a well defined geometry (representing points, lines, or polygons) in a reference space. Objects in GEO-RBAC correspond to sets of features of a given type.

A spatial role in GEO-RBAC represents a geographically bounded organizational function. The boundary can be defined as a feature, such as a road or a city. For example, a spatial role may specify that role taxi driver is defined for Milan or for Genoa. Roles are activated based on the position of the user. Besides a physical position, obtained from a given mobile terminal such as a GPS based vehicle tracking device or a cellular phone, users are also assigned a logical and device independent position, representing the feature in which the user is located. Logical positions can be computed from real ones by using specific mapping functions. To make the model more flexible, we assume that logical positions can be represented at different granularities, depending on the spatial role played by the user. To specify the type of the spatial boundary of the role and the granularity of the logical position, we introduce the concept of spatial role schema. Spatial roles are thus specified as instances of role schemas. The usage of role schemas and instances makes GEO-RBAC quite flexible since the type of role extents and logical positions can be customized (and the definition re-used), depending on the function the role represents. In conclusion, the main contributions of the paper are twofold: the proposal of a comprehensive framework for dealing with the spatial content of an access control system; the extension of the RBAC framework with the concept of role schema/instance, thus of a meta-level for roles that, if properly adapted, can be applied beyond the spatial context.

In this paper, we first present GEO-RBAC as an extension of the flat RBAC model. Then, we discuss how GEO-RBAC can be extended to deal with hierarchies. More precisely, the paper is organized as follows. In Section 2, we discuss related work. The reference geometric model we consider in this paper and its usage in GEO-RBAC are introduced in Section 3. In Section 4, we present the core model of GEO-RBAC whereas hierarchies are discussed in Section 5. An overall example is then presented in Section 6. Finally, Section 7 presents some concluding remarks and outlines future work.

2. RELATED WORK

In GIS (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 [1, 4] and 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 [3]. The model introduces the concept of spatial authorization as an authorization 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 authorization 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 multigranularity of spatial data. A similar architecture, but focused on XML-based representation of spatial data, has been proposed in [15]. A more complex spatial data model has been assumed in [2]. In this work, an access control system is presented that allows the specification of authorization rules to access complex structured spatial data stored in a DBMS and organized 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 [10, 11], 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 [6, 7, 8, 16]. Of particular interest is the access control system proposed in [6, 7], 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 roles but also protected objects.

3. SPATIAL INFORMATION IN GEO-RBAC

In order to make RBAC spatially aware, we need to first introduce the reference geometric model we want to use. In GEO-RBAC, the geometric model is used to represent objects, to model user positions, and to assign spatial extents to roles.

3.1 The reference geometric model

The geometric model describes how locations on Earth are represented in GEO-RBAC. We assume objects to be embedded in the Euclidean space E whilst a spatial reference system maps locations in E onto places on Earth. We assume objects to have a geometric representation (geometry) compliant with the OGC (Open GeoSpatial Consortium) *simple feature* geometric model [12]. We adopt this model because it is widely deployed in commercial spatial DBMSs and GISs. Although a more advanced spatial data model has been recently proposed [13, 14], we do not loose in generality by adopting the simple feature model.

In such a model, the geometry of an object can be of type point, line or polygon, or recursively be a collection of disjoint geometries. A point describes a single location in the coordinate space; a line represents a linear interpolation of an ordered sequence of points; a polygon is defined as an ordered sequence of closed lines defining the exterior and interior boundaries of an area. An interior boundary defines a hole in the polygon.

In GEO-RBAC, we consider the set of all geometries contained in a reference space (a polygon) and we denote it with GEO. We denote with MBB the reference space.

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 et al. in [5]. To exemplify, the $Contains(x, y)$ relationship between geometries x and y holds when all points of y are also points of x.

3.2 Spatially aware objects

We assume that resources to be protected consist of data about entities of the real world that may occupy a position. To be compliant with the OGC terminology, we call these entities *features* [12]. Features are identified by names. $Milan$, lake $Michigan$, car identified by $AZ213JW$ are examples of features. Features are *spatial* when entities can be mapped onto locations in the given space (for example, $Milan$ and lake $Michigan$). The location of a feature is represented through a geometry. Conversely, features are *non-spatial* when they are not associated with any location (for example car identified by $AZ213JW$). The sets of spatial features and non-spatial features are denoted in the following respectively by F_s and F_{ns} with $F_s \cap F_{ns} = \emptyset$. We define the set of features $F = F_s \cup F_{ns}$. Feature location is formally defined as follows.

DEFINITION 1 (FEATURE LOCATION). *Let F be the set of features and GEO be the set of geometries in space E. Feature location is a function $LocObj : F \rightarrow GEO \cup \{\perp\}$. Given a feature $f \in F$, the location $LocObj(f)$ is either a geometry in GEO if $f \in F_s$ or undefined ($\perp$) if $f \in F_{ns}$. We assume that the dimension of a feature f, denoted by $dim(f)$, is the geometric type of its location: 0 if it is a point, 1 if it is a line, 2 if it is a polygon, and $\perp$ if $f \in F_{ns}$.*

Features have an application dependent semantics that is expressed through the concept of *feature type* [12]. A feature type captures the intensional meaning of the entity. $Road$, $Town$, $Lake$, Car are examples of feature types. The *extension* of a feature type ft, denoted by $ext(ft)$, is a set of semantically homogeneous features. We assume, without loosing in generality, that the dimension of a spatial feature type is the dimension of its instances. For example, $Road$ may have dimension 1 whereas $Town$ and $Lake$ may have dimension 2. A feature type is instead non-spatial when the extension only includes non-spatial features (for example, Car). The two sets of spatial feature types and non-spatial feature types are indicated respectively by FT_s and FT_{ns} with $FT_s \cap FT_{ns} = \emptyset$. We define the set of features types $FT = FT_s \cup FT_{ns}$. Next definition introduces some functions relevant for feature management.

DEFINITION 2 (FEATURES FUNCTIONS). *Let $FT = \{ft_1, \ldots, ft_n\}$ be the set of feature types, F and GEO the set of features and geometries, respectively. We define:*

- *$FT\_dim : FT \rightarrow \{0, 1, 2, \perp\}$ such that, given a feature type ft, $FT\_dim(ft) = 0$ if ft is of type point, $FT\_dim(ft) = 1$ if ft is of type line, $FT\_dim(ft) = 2$ if ft is of type polygon, $FT\_dim(ft) = \perp$ when $ft \in FT_{ns}$.*
- *$Ext : FT \rightarrow 2^F$, the mapping from a feature type, either spatial or non-spatial, to a subset of features such that, given $ft_i \in FT$, $\forall f \in Ext(ft_i)$, $dim(f) = FT\_dim(ft_i)$. Given a feature type ft_i, Ext(ft_i) represents the* extension *of ft_i.*
- *$FT\_Type : F \rightarrow FT$ the mapping from a feature to its feature type.*

In some application contexts, it may happen that some spatial relationship exists between feature type extensions, defining a partial order between feature types. Consider for example feature types $Region$ and $Town$. It is reasonable to assume that the geometry associated with instances of $Town$ is contained in the geometry of instances of $Region$. As we will see, such relationship will be useful in characterizing the relationships between locations and role extents.

DEFINITION 3 (FEATURE TYPE CONTAINMENT). *Let $ft_i \in FT, ft_j \in FT, i \neq j$. We say that ft_i is contained in ft_j, denoted by $ft_i \subseteq_{ft} ft_j$, if $\forall f_i \in Ext(ft_i)\ \exists f_j \in Ext(ft_j)$ such that $LocObj(f_i) \subseteq LocObj(f_j)$.*

In order to more easily assign permissions, we assume that objects in GEO-RBAC are represented as subsets of feature type extensions. Formally, objects are defined as follows.

DEFINITION 4 (OBJECTS IN GEO-RBAC). *Let FT the set of feature types. Objects in GEO-RBAC are defined as $OBJ = \bigcup_{ft \in FT} 2^{Ext(ft)}$. Thus, the set OBJ consists of all possible subsets of feature type extensions.*

Objects in GEO-RBAC can be extensionally represented by listing the features belonging to the set or by intensionally specifying a query either spatial or non-spatial over a feature type extension. The object in this case corresponds to the query result.

3.3 Spatial role

The central notion in GEO-RBAC is that of *spatial role* defined as a pair $< r, e >$, where r is the role name and e the *spatial extent* (extent for short) of the role. The role extent defines the boundaries of the space in which the role can be assumed by the user.

Moreover, it seems reasonable to assume that the extent of a role, besides a geometry, has a semantic characterization. Thus, we assume role extents to be modelled as features of possibly different feature types. As a further consideration, note that in real applications it makes sense to have also non-spatial roles. For example, it does not seem reasonable to assign spatial extents to roles related to company organizations such as $Manager$ or $Employee$. However, for the sake of uniformity, we consider non-spatial roles to be a subset of spatial roles having the reference space, i.e. MBB, as role extent.

DEFINITION 5 (ROLE EXTENT). *Let R be a set of role names, let $REXT\_FT \subseteq FT$ be the set of role extent feature types. The set of role extents, denoted by $REXT$, is defined as $REXT = \bigcup_{ft \in REXT\_FT} Ext(ft) \cup \{MBB\}$.*

Notice that the same role name can appear in different spatial roles. For example the role *Driver* can be associated with different extents, say the city of Milan or Rome, to form distinct spatial roles.

3.4 Position Model

In GEO-RBAC, we assume users to have a position that can change in time. Positions can be real or logical. The real position corresponds to the position on the Earth of the user, obtained from a given mobile terminal such as a GPS based vehicle tracking device or a cellular phone. Real positions can be represented as geometries of different types since, depending on the chosen technology and accuracy requirements, they may correspond to points or polygons. For the sake of generality we do not make any assumption on the geometric type of the real position.

Besides real positions, however, for activating a given role, it may be useful to know not only the real position of the user but also the logical one. The logical position allows a position to be represented in a way that is almost independent from the underlying positioning technology. Logical position is modelled as a spatial feature. For example the logical location of a vehicle may be a polygonal feature of type, say, $city$. Such a feature can already exist in the information base or be a new feature entered into the system when the position is notified. Positions can also be represented at varying granularity levels which may depend on the role played by the user: for example for a taxi driver the logical position can be a point along a road while for a truck driver it may be a portion of road. Note that a coarse position may be requested for privacy-preserving purpose, in order to hide the actual position of user.

The logical position can be computed from real positions by using specific mapping functions.

For example, a function could be defined to map a point acquired through GPS based equipment onto the closer road segment.

DEFINITION 6 (POSITIONS). *The set $RPOS$ of real positions is a subset of geometries in GEO, thus $RPOS \subseteq GEO$. The set $LPOS$ of logical positions consists of features of type in $LPOS\_FT \subseteq FT$, thus $LPOS$ is defined as $LPOS = \bigcup_{ft \in LPOS\_FT} Ext(ft)$.*

Given a feature type ft, we call position mapping function for *ft a function m_{ft} defined as $m_{ft} : RPOS \rightarrow LPOS$ such that $m_{ft}(rp) = f$ and $f \in Ext(ft)$. The function m_{ft}, given a real position rp, returns a logical position corresponding to an instance of ft having rp as real position.*

A position mapping function is a total function, thus the logical position can be computed for any real position. Moreover, we assume to have at least one position mapping function for each feature type ft. We denote with M the set of all position mapping functions.

Notation	Meaning
FT	Feature types
F	Features
R	Role names
$REXT\_FT$	Feature types of role extents
$LPOS\_FT$	Feature types of logical positions
$REXT$	Role extents
$LPOS$	Logical positions
$RPOS$	Real positions
M	Position mapping functions
OBJ	Objects

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

4. THE GEO-RBAC CORE MODEL

The central idea of GEO-RBAC is the distinction between the concept of role schema and role instance (or spatial role). A role schema defines some common properties of a set of spatially aware organizational functions with a similar meaning. A role schema not only defines a common name for a set of spatial roles but also constrains the space where roles can be enabled. Moreover it specifies the type of logical locations and ultimately the granularity of the position that the users playing that role may occupy. A role instance is a role fulfilling the constraints defined at schema level. A spatial role has thus the same name of the schema role name whereas the spatial boundary of the role is a spatial feature with a precise semantics. It should be noticed that all spatial roles instantiating a role schema are fully identified by the role extent (feature) name. Another important property of the role schema is that it may be assigned permissions. Those permissions are then inherited and shared by all the instances of the role schema.

Users are assigned spatial roles, thus instances of some role schema that can be activated during a session. Unlike RBAC, roles are *enabled* only when the user position is contained in the role extent.

For sake of readability, in what follows we present the model as organized in a number of logical parts, one for each major set of the RBAC model, i.e. roles, permissions, users, sessions. The general structure of the model is illustrated in Figure 1. We use the graphical representation adopted in RBAC. In defining the model, we refer to the notation introduced in the previous section and summarized in Table 1.

4.1 Role schemas and instances

A role schema defines a common name for a set of roles, the feature type of the role extent, the feature type of the logical locations and the mapping function relating real positions with logical positions.

DEFINITION 7 (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 $< TaxiDriver, RoadNetwork, PointOnRoad, m_{PointOnRoad} >$ in which: $TaxiDriver$ is the name of the role; $RoadNetwork$ the feature type of the role extent, thus the "kind of object" that spatially constrains the role; $PointOnRoad$ the feature type of logical positions consisting of points along road lines; finally $m_{PointOnRoad}$ the position mapping function that maps a real position into a logical one. Such a function might compute the point on road closer to the real position of the user.

From Definition 7 it follows that the feature type representing the logical location must precede in the ordering the feature type representing the role extent. According to Definition 3, this means that logical positions must be contained in role extents. Thus, it cannot occur that a location only partially overlaps the space defined by the role extent. In the above example, the feature type $PointOnRoad$ precedes $RoadNetwork$ since we assume all intances of $PointOnRoad$ are points contained in the lines representing roads. From this assumption it follows that it is always possible to determine whether the logical location of a user is contained in a role extent and thus which roles in the session are enabled.

Based on the previous definition, the role schema for a role name r is unique. This means that different schemas for the same role, such as: $< TaxiDriver, RoadNetwork, loc, m_{loc} >$ and $< TaxiDriver, Region, loc_1, m_{loc_1} >$ are not allowed. Should the application require a role on different types of extents, a hierarchy of role schemas has to be defined (see Section 5).

Given a role schema, role instances can be simply created by specifying for the role name its extent as a feature of the type specified in the schema.

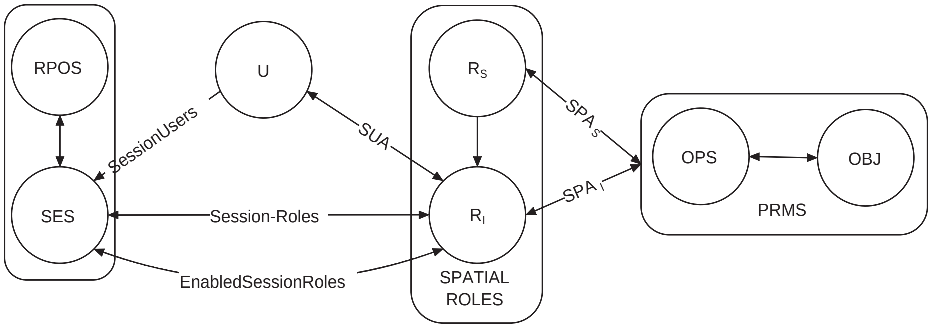

Figure 1: Core GEO-RBAC

Notice that of the four components of a role schema, only the first two are actually needed for the specification of a role instance. As we will see, the last two components, involving the notion of logical position, are needed for role activation. To indicate the component α of role schema r_s, we use the notation $r_s.\alpha$.

Definition 8 (Role Instance). *Given a role schema r_s, an instance r_i of r_s is a pair $< r, e >$ where $r = r_s.r$ and $e \in F$, such that $FT\_Type(e) = r_s.ext$. The schema of r_i is denoted by $SchemaOf(r_i)$.*

We denote with $R_I \subseteq R \times REXT$ the set of role instances for all role schemas. For the sake of readability, a role instance $< r, e >$ is also denoted by $r(e)$.

4.2 Permissions

In GEO-RBAC, permissions can be associated either with the role schema and inherited by all role instances of the schema or directly with role instances. Such different granularities are formalized by introducing two functions: $S\_PrmsAssignment$, relating roles schemas and permissions sets; $I\_PrmsAssignment$ relating spatial roles, thus role instances, to specific permissions. Function $I\_PrmsAssignment^*$ is then introduced to combine permissions directly assigned to spatial roles with permissions inherited from their role schema.

Definition 9 (Permissions). *Let R_S be the set of role schemas, R_I the set of role instances, OPS the set of operations, OBJ the set of objects. The set of permissions $PRMS$ is defined as $PRMS = 2^{(OPS \times OBS)}$. 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 schemas onto sets of permissions. Given a role schema r_s, $S\_PrmsAssignment(r_s) = \{p \in PRMS| < r_s, p > \in SPA_S\}$;*
- *$SPA_I : R_I \times PRMS$, a many-to-many mapping permission-to-spatial role instance assignment relation;*
- *$I\_PrmsAssignment : R_I \rightarrow 2^{PRMS}$ the mapping of spatial role instances onto sets of permissions. Given a role instance r_i, $I\_PrmsAssignment(r_i) = \{p \in PRMS| < r_i, p > \in SPA_I\}$;*
- *$I\_PrmsAssignment^* : R_I \rightarrow 2^{PRMS}$ such that given a role instance r_i, $I\_PrmsAssignment^*(r_i) = I\_PrmsAssignment(r_i) \cup S\_PrmsAssignment(SchemaOf(r_i))\}$.*

Hence the permissions of a role are those assigned to its schema plus those directly assigned to the instance.

4.3 Users

Spatial roles are assigned to users. The definition of the model for this part is conceptually analogous to that in RBAC.

Definition 10 (Users). *Let U be the set of users and R_I be the set of role instances. We define:*

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

4.4 Sessions

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. However, for a session role to be enabled, the user should be logically located within the space of the role extent. 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*.

Definition 11 (Sessions). *Let U be the set of users and SES the set of 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 $SessionRoles(s) \subseteq \{< r, e > \in R_I | (SessionUser(s), < r, e >) \in SUA\}$. real position of the session user.*

$SessionRoles(s)$ corresponds to the roles that can be potentially activated in session s. However, depending on the user position during that session, only a subset of such roles is enabled and permissions granted. To determine enabled roles, containment between logical user position and role extent has to be assessed. Then, for each enabled role, the set of permissions assigned to the corresponding role schema is determined.

Definition 12 (Enabled Roles). *Enabled session roles are defined as the function:*

$EnabledSessionRoles : SES \times RPOS \rightarrow 2^{R_I}$ *such that* $EnabledSessionRole(s, rp) = \{< r, e >\in R_I| < r, e >\in SessionRoles(s),\ lpos = SchemaOf(r).m_{loc}(rp), Contains(LocObj(e), LocObj(lpos)) = TRUE\}$.

Enabled roles are the basis for determining whether to grant or reject an access request, i.e., for defining the authorization control mechanism. An access request is a tuple $\langle s, rp, p, o\rangle$ stating that the user of session s in position rp wants to perform operation p on object o, thus $\langle s, rp, p, o\rangle \in SES \times RPOS \times OPS \times OBJ$. An access request can be satisfied at real position rp, if permission (p, o) belongs to the set of permissions assigned to the roles that are enabled in s when the session user is in position rp.

DEFINITION 13 (AUTHORIZATION CONTROL FUNCTION). *An access request is a tuple* $ar = \langle s, rp, p, o\rangle \in SES \times RPOS \times OPS \times OBJ$. ar *can be satisfied at position* rp *if*

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

5. HIERARCHIES IN GEO-RBAC

As Hierarchical RBAC adds to Flat RBAC the support for role hierarchies [17], hierarchical GEO-RBAC (GEO-HRAC) adds to GEO-RBAC the support to model hierarchies. According to [17], the hierarchical level can be defined by introducing a partial order $\preceq$ between roles such that $r_i \preceq r_j$ means that: (i) r_j inherits all permissions assigned to r_i; (ii) users which have been assigned r_j have also assigned r_i.

Moreover, since in GEO-RBAC we introduce the concept of enabled role, we also assume that (iii) if r_j is enabled, and thus the user can play that role in a session s, also r_i results to be enabled in s.

Since in GEO-RBAC permissions can be assigned both at the schema and the instance level, two different hierarchies can be defined as illustrated in Figure 2. At the schema level, the hierarchy allows us to define a partial order $\preceq_s$ between role schemas. Such a hierarchy is then inherited at the instance level.

At the schema level, similarly to HRBAC, the partial order is defined according to the semantics of the considered application domain. Given two role schemas r_{s_1} and r_{s_2}, if $r_{s_1} \preceq_s r_{s_2}$ then r_{s_2} inherits all the permissions of r_{s_1}. We assume that such ordering can be defined only when containment holds between the extent and the location types of the role schemas. For example, given the schemas:
$citizen =< Citizen, City, PointInCity, m_{PointInCity>}$
$taxiDriver =< TaxiDriver, UrbanRoadNetwork,$
$PointOnRoad, m_{PointOnRoad>}$
$Citizen \preceq_s TaxiDriver$ means that taxi drivers have at least the same permissions of citizens.

At the instance level, for how the model is defined, all the role instances inherit the permissions assigned to their role schema and thus also the permissions of the inherited roles. Therefore the role instance $TaxiDriver(MilanRoad)$ will also inherit the permissions of both the taxi driver and the citizen role schema. Moreover, if $r_1(g_1)$ and $r_2(g_2)$ are two instances of schemas r_1 and r_2 such that $r_1 \preceq_s r_2$ then $r_1(g_1) \preceq_i r_2(g_2)$ means that not only $r_2(g_2)$ inherits the permissions of the role schema of r_1 but also the permissions that have been assigned specifically to the instance $r_1(g_1)$. Suppose the role instance $Citizen(Milan)$ has been given a specific permission. Then $Citizen(Milan) \preceq_i TaxiDriver(MilanRoad)$ means that the taxi driver will also inherit the permissions of the Milan citizen.

Another case to be considered is when the roles are instances of the same role schema. Consider the instances $TaxiDriver(MilanRoad)$ and $TaxiDriver(RoadCentreMilan)$ with the geometry of $RoadCentreMilan$ (describing only the roads in the downtown of Milan) contained in $MilanRoad$ (all roads in Milan). Since we assume $RoadCentreMilan$ to be contained in $MilanRoad$, a taxi driver with spatial extent the centre of Milan necessarily inherits all the permissions of the taxi driver with the larger extent. In this case, we consider this hierarchy be implicitly defined.

To summarize, GEO-HRBAC can be formally defined as follows.

DEFINITION 14 (GEO-HRBAC). *GEO-HRBAC is defined from GEO-RBAC by introducing a partial order between role schemas and instances. We define the hierarchy at the schema level as follows:*

1. $RH_s \subseteq R_S \times R_S$, *a partial order over* R_S, *denoted by* $\preceq_s$. *If* $r_{s_1} \preceq_s r_{s_2}$ *holds, we assume that* $r_{s_2}.ext \subseteq_{ft} r_{s_1}.ext$, *and* $r_{s_2}.loc \subseteq_{ft} r_{s_1}.loc$;
2. $S\_AuthorizedPrms : R_S \rightarrow 2^{PRMS}$ *such that, given a role schema* r_s, $S\_AuthorizedPrms(r_s)$ *returns all permissions assigned to* r_s *and to all its ancestors, i.e.* $S\_AuthorizedPrms(r_s) = \{p \in PRMS|r'_s \preceq_s r_s, < r'_s, p >\in SPA_S\}$.

We define the hierarchy at the instance level as follows:

1. $RH_i \subseteq R_I \times R_I$, *a partial order over* R_I, *denoted by* $\preceq_i$. $< r_1, e_1 >\preceq_i< r_2, e_2 >$ *holds if* $SchemaOf(< r_1, e_1 >) \preceq_s SchemaOf(< r_2, e_2 >)$ *and* $LocObj(e_2) \subseteq LocObj(e_1)$.
2. $I\_AuthorizedPrms : R_I \rightarrow 2^{PRMS}$ *such that, given a role instance* r_i, $I\_AuthorizedPrms(r_i)$ *returns all permissions assigned to* r_i *and to all its ancestors, i.e.,* $I\_AuthorizedPrms(r_i) = \{p \in PRMS|r'_i \preceq_i r_i, p \in I\_PrmsAssignment^*(ri')\}$.
3. $I\_AuthorizedUsers : R_I \rightarrow 2^U$ *such that, given a role instance* r_i, $I\_AuthorizedUsers(r_i)$ *returns all users assigned to* r_i *and to all its descendants, i.e.,* $I\_AuthorizedUsers(r_i) = \{u \in U|r_i \preceq_i r'_i, < u, r'_i >\in SUA\}$.

From the previous definition it follows that the ordering between role schemas corresponds to the ordering of position granularities: the location becomes more precise as the role becomes more specific while the extension gets smaller. That is like to say that the more "powerful" roles are those operating on smaller regions. Moreover, we note that the ordering between instances of the same schema (i.e., spatial roles with the same name) is implicitly defined by the containment relationship existing between their extents.

Based on the above definition, it is possible to show a number of properties of role hierarchies. Some of them derive from the permission and user inheritance and coincide with properties that hold also for HRBAC. However, based on the containment relationship existing between spatial roles, a new property can be stated concerning enabled roles. Assume that a role r_{i_2} is enabled in a session s and a real position rp and that $r_{i_1} \preceq_i r_{i_2}$. Since from the definition of $\preceq_i$ the spatial extent of r_{i_2} is contained in the spatial extent of r_{i_1}, this means that also r_{i_1} is enabled. In our example, this means that if $TaxiDriver(MilanRoad) \preceq_i TaxiDriver(RoadCentreMilan)$, when $TaxiDriver(RoadCentreMilan)$ is enabled in a certain position, also $Taxi\_driver(MilanRoad)$ is enabled. The proof of the following proposition trivially follows from Definition 14.

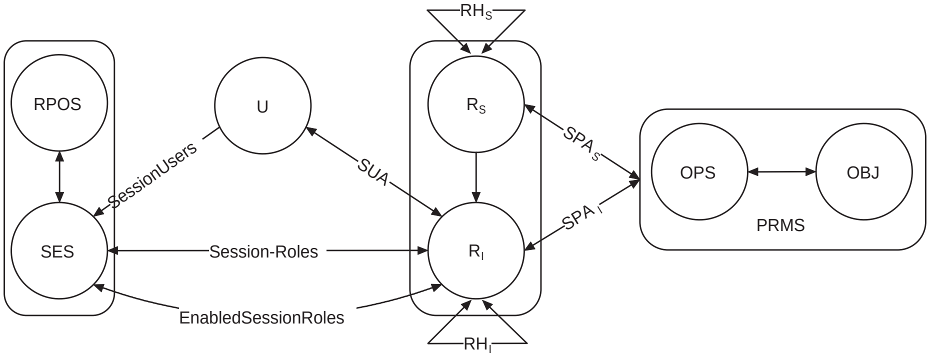

Figure 2: Hierarchical GEO-RBAC

Basic objects
$FT = \{UrbanRoadNetwork, Accident, City, AreaInCity, Monument, PointOnRoad\}$
$OBJ = \{Ext(UrbanRoadNetwork), Ext(Accident), Ext(Monument)\}$
$OPS = \{GetTrafficInfo, Notify, Find\}$
$PRMS = \{p_1, p_2, p_3\}$ with $\begin{cases} p_1 = (GetTrafficInfo, Ext(UrbanRoadNetwork)) \\ p_2 = (Notify, Ext(Accident)) \\ p_3 = (Find, Ext(Monument)) \end{cases}$

Schema
$R = \{Citizen, TaxiDriver, Tourist\}$
$REXT\_FT = \{City, UrbanRoadNetwork, AreaInCity\}$
$LPOS\_FT = \{PointOnRoad\}$
$R_S = \{r_{s_1}, r_{s_2}, r_{s_3}\}$ with $\begin{cases} r_{s_1} =< Citizen, City, PointOnRoad, m_{PointOnRoad} > \\ r_{s_2} =< TaxiDriver, UrbanRoadNetwork, PointOnRoad, m_{PointOnRoad} > \\ r_{s_3} =< Tourist, AreaInCity, PointOnRoad, m_{PointOnRoad} > \end{cases}$

Instances
$REXT = \{Milan, CentreMilan, RoadMilan\}$
$R_I = \{Citizen(Milan), TaxiDriver(RoadMilan), Tourist(CentreMilan\}$

Schema role hierachy
$r_{s_1} \preceq_s r_{s_2} \quad r_{s_1} \preceq_s r_{s_3}$

User assignment
$U = \{John, Paul\}$
$SUA = \{s_{ua_1}, s_{ua_2}, s_{ua_3}, s_{ua_4}\}$ with $\begin{cases} s_{ua_1} = \langle John, Citizen(Milan)\rangle \\ s_{ua_2} = \langle Paul, Citizen(Milan)\rangle \\ s_{ua_3} = \langle John, TaxiDriver(RoadMilan)\rangle \\ s_{ua_4} = \langle Paul, Tourist(CentreMilan)\rangle \end{cases}$

Permission assignment
$S\_PrmsAssignement(r_{s_1)} = \{p_1\} \quad S\_PrmsAssignement(r_{s_2)} = \{p_1, p_2\} \quad S\_PrmsAssignement(r_{s_3}) = \{p_1, p_3\}$

Sessions
$SES = \{s_1, s_2\} \quad UserSession(s_1) = \{John\} \quad UserSession(s_2) = \{Paul\}$

EnabledRoles
$EnabledSessionRoles(s_1, loc_1) = \{TaxiDriver(RoadMilan), Citizen(Milan)\}$ if $m_{PointOnRoad}(loc_1)$ is in $RoadMilan$
$EnabledSessionRoles(s_2, loc_2) = \{Citizen(Milan)\}$, if $m_{PointOnRoad}(loc_2)$ is not in the centre of Milan

Figure 3: An example of a GEO-RBAC application

PROPOSITION 1. *Let $r_{s_1} \in R_S, r_{s_2} \in R_S$, Let $r_{i_1} \in R_I, r_{i_2} \in R_I$, such that $SchemaOf(r_{i_1}) = r_{s_1}$ and $SchemaOf(r_{i_2}) = r_{s_2}$. Suppose that $r_{s_1} \preceq_s r_{s_2}$ and $r_{i_1} \preceq_i r_{i_2}$. The following properties hold:*

- $S\_AuthorizedPrms(r_{s_1}) \subseteq S\_AuthorizedPrms(r_{s_2})$;
- $I\_AuthorizedPrms(r_{i_1}) \subseteq I\_AuthorizedPrms(r_{i_2})$;
- $I\_AuthorizedUsers(r_{i_2}) \subseteq I\_AuthorizedUsers(r_{i_1})$;
- *for all $s \in SES, rp \in RPOS$, $r_{i_2} \in EnabledSessionRole(s, rp)$ implies that $r_{i_1} \in EnabledSessionRole(s, rp)$.*

6. A COMPREHENSIVE EXAMPLE

To finally summarize the characteristics of the model, we discuss an extended example. Consider the scenario introduced in Section 1, concerning an information service providing traffic information. Suppose that the considered feature types are $\{UrbanRoadNetwork, Accident, City, AreaInCity, Monument, PointOnRoad, RoadSegment\}$ and suppose the considered objects correspond to the extentions of feature types $UrbanRoadNetwork$, $Accident$, $Monument$. Assume, moreover, that permissions are defined to receive traffic information concerning roads ($GetTrafficInfo$ operation over $UrbanRoadNetwork$ features), to be notified in case of accident ($Notify$ operation over $Accident$ features) and to locate monuments ($Find$ operation over $Monument$). In this scenario, we consider the roles: $Citizen$, $TaxiDriver$ and $Tourist$. The role schema of, say, $Citizen$ can be modelled as follows: $< Citizen, City, PointOnRoad, m_{PointOnRoad} >$, where $m_{PointOnRoad}$ is a functions mapping real user positions in $PointOnRoad$ features. The schemas are hierarchically ordered in such a way that taxi drivers and tourists are also citizens. The unique permission assigned to citizens is for getting traffic information on roads. This permission is thus inherited by taxi drivers, that in addition can also be notified of accidents, and also by tourists, that in addition are allowed to locate monuments in an area of the city. We define the following role instances: $Citizen(Milan)$, $TaxiDriver(RoadMilan)$ and $Tourist(CentreMilan)$.

Now, consider two users, say $John$ and $Paul$. $John$ is a taxi driver in Milan whereas $Paul$ is a tourist visiting the centre of Milan. Suppose that $John$ starts a session s_1 in (real) position loc_1 and $Paul$ a session s_2 in (real) position loc_2. If the $PointOnRoad$ logically corresponding to loc_1 is contained in $RoadMilan$, then role $TaxiDriver(RoadMilan)$ is enabled for $John$ during session s_1, otherwise it is not. Similarly, if the $PointOnRoad$ logically corresponding to loc_2 is contained in $CentreMilan$, role $Tourist(CentreMilan)$ is enabled for $Paul$ during session s_2. Conversely the only enabled role for $Paul$, assuming that Paul is anyway in Milan, is that of citizen.

7. CONCLUSIONS

In this paper we have presented GEO-RBAC, an extension of the RBAC model dealing with spatial and location-based information. Unlike other proposals of spatially aware access control models, GEO-RBAC relies on the OGC spatial model [12] to model (spatial) objects, user positions, and geographically bounded roles, making the approach quite standard and flexible. Another important characteristic of the model is the ability to deal with either real positions, obtained from a given mobile terminal or a cellular phone, and logical ones, possibly represented at different granularities. By introducing the concept of role schema, the type of role extents and logical positions can be customized, depending on the function the role represents. Moreover, besides the concept of active role, the concept of enabled role has also been introduced, in order to determine the roles that are enabled in sessions based on user positions. Finally, like RBAC, GEO-RBAC has then been extended with 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. Future work includes the definition of constraints for the model. The idea is to extend the RBAC constraints to deal with conflicting role extents and user positions. An additional issue concerns the XML-based representation of the access control model. The prospected approach is to base the XML represention of the GEO-RBAC model on GML (Geography Markup Language) [14] for all spatial aspects. We also plan to extend the model for use in sensor-based applications and pervasive computing environments. Finally, another important topic to consider is the authenticity of location information.

8. ACKNOWLEDGMENTS

The work of B.Catania and M.Damiani has been partially supported by the Italian MIUR under the project *Web-based management and representation of spatial and geographical data*.

9. REFERENCES

[1] V. Atluri and P. Mazzoleni. A Uniform Indexing Scheme for Geo-spatial Data and Authorizations. In *Proc. of the Sixteenth Conf. on Data and Application Security, IFIP TC11/WG11.3*, Cambridge, UK, 2002, pages 207–218.

[2] A. Belussi, E. Bertino, B. Catania, M.L. Damiani, and A. Nucita. An Authorization Model for Geographical Maps. In *Proc. of the 12th International Symposium of ACM GIS*, Washington DC ,USA, November 2004, pages 82–91.

[3] E. Bertino, M.L. Damiani, and D. Momini. An Access Control System for a Web Map Management Service. In *Proc. of the 14th International Workshop on Research Issues in Data Engineering (RIDE-WS-ECEG)*, Boston, USA, March 2004, pages 33–39.

[4] S.A. Chun and V. Atluri. Protecting Privacy from Continuous High-resolution Satellite Surveillance. In *Proc. of the 14th IFIP 11.3 Working Conference on Database Security*, August 2000, pages 233–244.

[5] E. Clementini, P. di Felice, and P. van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *LNCS 692: Proc. of the 3yh International Symposium on Advances in Spatial Databases SSD'93*, June 1993, pages 277–295.

[6] M. Covington, W. Long, S. Srinivasan, A.K. Dev, M. Ahamad, and G.D. Abowd. Securing Context-aware Applications Using Environment Roles. In *Proc. of the 6th ACM Symposium on Access Control Models and Technologies*, Chantilly, Virginia, USA, 2001, pages 10–20.

[7] M. Covington, M. Moyer, and M. Ahamad. Generalized Role-based Access Control for Securing Future Applications. In *Proc. of 23rd National Information Systems Security Conference*, Baltimore, USA, October 2000.

[8] F. Cuppens and A. Miège. Modelling Contexts in the Or-bac Model. In *Proc. of 19th Annual Computer Security Applications Conference*. IEEE Computer Society, 2003, pages 416–427.

[9] D. Ferraiolo, R. Sandhu, S. Gavrila , and R. Kuhn and R. Chandramouli. Proposed NIST Standard for Role-Based Access Control. In *Proc. of ACM Transactions on Information and System Security*, Vol.4, 2001, pages 224–274.

[10] F. Hansen and V. Oleshchuk. Spatial Role-based Access Control Model for Wireless Networks. In *Proc. of IEEE Vehicular Technology Conference VTC2003-Fall*, Orlando, USA, October 2003.

[11] F. Hansen and V. Oleshchuk. Srbac: A Spatial Role-based Access Control Model for Mobile Systems. In *Proc. of Nordsec 2003, Gjøvik, Norway*, October 2003, pages 129–141.

[12] OpenGIS Consortium. OpenGIS Simple Features Specification for SQL. Technical Report OGC 99-049, 1999.

[13] OpenGIS Consortium. The OpenGIS Abstract Specification. Topic 1: Feature Geometry (ISO 19107 Spatial Schema. OpenGIS Project Document Number 01-101, 2001.

[14] OpenGIS Consortium. OpenGIS Geography Markup Language (GML) Implementation Specification. Technical Report OGC 02-023r4, 2003.

[15] B. Purevjii, T. Amagasa, S. Imai, and Y. Kanamori. An Access Control Model for Geographic Data in an XML-based Framework. In *Proc. of the 2nd International Workshop on Information Systems Security (WOSIS)*, 2004, pages 251–260.

[16] G. Sampemane, P. Naldurg, and R.H. Campbell. Access Control for Active Spaces. In *Proc. of 18th Annual Computer Security Applications Conference*, IEEE Computer Society, 2002, pages 343–352.

[17] R. Sandhu, D. Ferraiolo, and R. Kuhn. The NIST Model for Role-Based Access Control: Towards a Unified Standard. In *Proc. of the 5th ACM Workshop on Role-Based Access Control*, Berlin, Germany, 2000, pages 47–63.