5

An Approach To Deriving Global Authorizations in Federated Database Systems

Silvana Castano
Universitá di Milano
Dipartimento di Scienze dell'Informazione, Universitá di Milano
Via Comelico 39/41, Milano, Italy.
Email: castano@dsi.unimi.it

Abstract

Global authorizations in federated database systems can be derived from local authorizations exported by component databases. This paper addresses problems related to the development of techniques for the analysis of local authorizations and for the construction of global authorizations where semantic correspondences between subjects in different component databases are identified on the basis of authorization compatibility. Abstraction of compatible authorizations is discussed to semi-automatically derive global authorizations that are consistent with the local ones.

Keywords

Federated database systems, Discretionary access control, Authorization compatibility, Authorization abstraction.

1 INTRODUCTION

A federated database system (or federation) is characterized by a number of component databases (CDB) which share part of their data, while preserving their local autonomy. In particular, in the so-called tightly coupled systems [She90], a federation administrator (FDBA) is responsible for managing the federation and for defining a *federated schema*. A federated schema is an integrated description of all data exported by CDBs of the federation, obtained by resolving possible semantic conflicts among data descriptions [Bri94,Ham93,Sie91].

A basic security requirement of federated systems is that the autonomy of CDBs must be taken into account for access control [Mor92]. This means that global accesses to objects of a federated schema must be authorized also by the involved CDBs, according to their local security policies. Two levels of authorization can be distinguished: a *global level*, where global requests of federated users are evaluated

P. Samarati et al. (eds.), *Database Security*

against global authorizations defined for the objects in the federated schema, and a *local level*, where local accesses involved in the global request must authorized by local systems.
Several discretionary models have been proposed in the literature for access control in federated database systems, offering different degrees of flexibility to enforce different security policies, and different authentication schemes to enforce different levels of local autonomy [Jon93]. According to "view-based" discretionary models, federated users are authorized to access export schemas or views of the federated schema, defined by the security administrator, taking into account also local access rights of the involved objects [Tem87,Wan87]. According to "propagation-based" discretionary models, global requests on protection objects in the federated schema first are evaluated at the global level and then are propagated to local component systems for further evaluation, exploiting properly defined mappings [Jon94].

Definition of global authorizations is an important activity to be performed in a federation. In particular, for objects of the federated schema that are derived from the integration of local objects in CDBs, it is important to derive global authorizations that properly integrate the corresponding local access rights, to assure a consistent authorization state in the federation. Issues related to the integration of secrecy features in a federated database system enforcing a mandatory security policy have been discussed in [Idr94].

In the paper, we address the problem of deriving global authorizations for the integrated objects of a federated schema, starting from local authorizations specified for the involved local objects in CDBs. We propose a semi-automatic approach, based on the analysis of local authorizations exported by CDBs, and on the abstraction of local authorizations that are "compatible". Authorization compatibility is evaluated using criteria and metrics based on a structured *dictionary*, where knowledge about the application domain of the federation and about schema integration is maintained, in form of names and semantic relationships between names. Global authorization obtained by means of abstraction are consistent with the corresponding local authorizations, that is, they specify privileges complying with local security requirements of different CDBs for local objects that have been integrated.

The paper is organized as follows. In Section 2, we introduce the basic concepts of the proposed approach. In Section 3, we illustrate proper criteria and associated metrics for analysis and comparison of authorizations in different CDBs. In Section 4, we describe subject clustering based on authorization compatibility. In Section 5, we describe the abstraction of global authorizations and, finally, in Section 6, we give our concluding remarks.

2 BASIC CONCEPTS OF THE APPROACH

In Fig. 1, we illustrate the steps of the approach we propose to semi-automatically deriving global authorizations that comply with schema integration. As we can see, derivation of global authorizations for objects of a federated schema is based on the analysis of local authorizations exported by each CDB_i for the objects of its component schema CS_i, and on proper abstraction of authorizations that are "compatible". The approach is intended to support the federation security

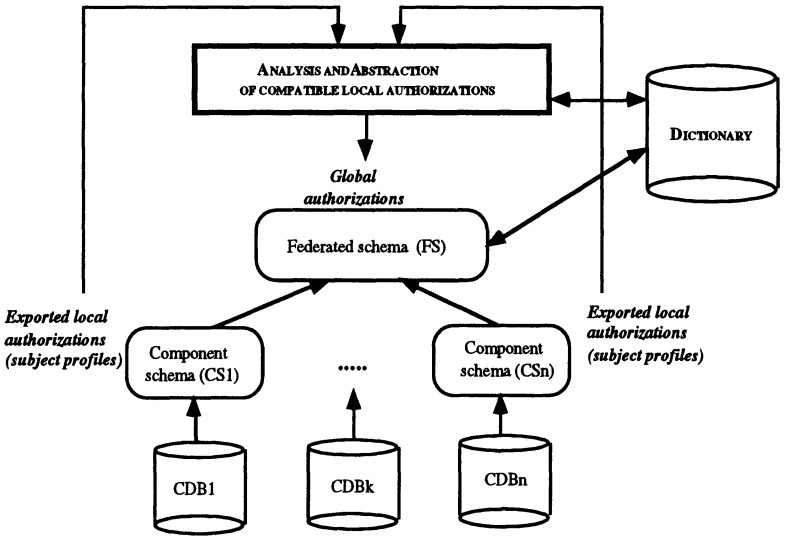

Figure 1 Overview of the approach

administrator in defining global authorizations in tightly coupled federations. If more than one federated schema is defined in the federation, the approach can be separately applied to each schema.

2.1 Protection objects in a federation

We refer to component schemas defined according to (or translated into) a common data model (e.g., object-oriented model, relational model) [She90]. To make the approach applicable to different types of models, we consider a schema composed of *objects* as the model-independent basis for representing concepts of the Universe of Discourse within a schema. An object corresponds to a relation in the relational model, or to an object class in an object-oriented model. An object is characterized by *structural properties* and *behavioral properties*. The former are used to describe static features of an object (e.g., attributes for both relations and object classes). The latter are used to describe dynamic features of an object (e.g., operations executable on relations, methods executable on object classes). Moreover, *links* describe relationships between objects (e.g., foreign keys of relations, implicit references of object classes). We consider a component schema CS_i of a component database CDB_i as a set of objects, $CS_i = \{o_{1i}, o_{2i}, \ldots, o_{ni}\}$, that we call *local objects*. Protection objects in a component schema are local objects and their structural properties.

In tightly coupled federations, schemas of component databases are integrated to derive a federated schema FS. During the schema integration process, all local objects that have the same real world semantics in different component schemas (i.e., semantically similar objects) are integrated into a unique object of FS. Main problems arising during schema integration are due to the fact that semantically

similar objects can have different representations in different local schemas, due to the fact that databases of a federation are generally designed separately, by different designers [Kim95]. Schema integration process consists of a *conflict analysis* phase and a *conflict resolution* phase. During the conflict analysis phase, possible conflicts (e.g., name conflicts, structural conflicts [Bat86]) that can arise between semantically similar objects are identified. During the conflict resolution phase, identified conflicts are properly resolved by selecting a reference representation [Kim95]. As the result of the integration, objects that are identical, synonyms, and compatible in different local schemas are integrated into a unique object in the federated schema, by properly merging their properties [Bat86]. For example, let us consider in the banking domain, a federation of databases related to different offices of the same bank, located in different countries (for the sake of simplicity, in the following we consider two component databases CDB_1 and CDB_2). As a simple example of object integration for this federation, let us consider in Fig. 2 a local object `Account` belonging to the schema CS_1 of CDB_1 and a second local object `Accounts` belonging to the schema CS_2 of CDB_2. These semantically similar objects describe data about accounts and a name conflict exist between them. Consequently, they are integrated into a global object named `Account` in the federated schema. Structural and behavioral properties of global object `Account` are the union of the corresponding properties of local objects `Account` and `Accounts`, and the set of its instances is the union of the corresponding instances of `Account` and `Accounts`. Moreover, the structural property `Classification:integer` has a null value for account instances of CDB_1. Issues related to the integration of component schemas are not further discussed here. A detailed classification of conflicts that can arise in a federation of object-oriented and relational databases together with corresponding resolution techniques is presented in [Kim95], while a survey of schema integration techniques is presented in [Bat86].

As the result of the schema integration process, a federated schema $FS = \{\overline{o}_1, \overline{o}_2, \ldots, \overline{o}_m\}$ is obtained whose objects are called *global objects*. For security purposes, global objects can be classified as follows:

1. *Integrated objects*: they are objects $\overline{o}_j \in FS$ obtained by integration local objects that are semantically similar in different component schemas.
2. *Local objects*: they are objects $\overline{o}_j \in FS$ defined in some component schema of the federation, which are imported "as-is" in FS, without modifications / integration with other objects.
3. *Federated objects*: they are objects $\overline{o}_j \in FS$ defined by the FDBA on the basis of specific requirements of the applications of the federation. Federated objects do not have corresponding local objects in component schemas of the federation and are stored in a special database, different from federation's CDBs.
4. *Composite objects*: they are objects $\overline{o}_j \in FS$ obtained by aggregation of a number of objects of FS belonging to any of the previous categories. Like federated objects, also composite objects are defined by the FDBA to fulfill specific requirements of the applications of the federation.

In this paper, we focus on the definition of global authorizations for the integrated objects of a federated schema, with the aim of deriving global authorizations consis-

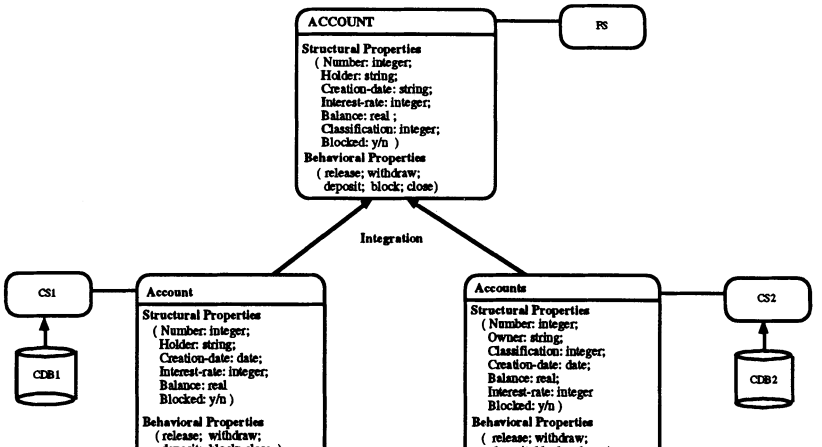

Figure 2 Example of object integration

tent with local ones exported by federation's CDBs for corresponding local objects. For example, with reference to Fig. 2, let us suppose to have: i) two local authorizations exported by CDB_1, one to perform a `read` operation on property `Balance` and the other to perform a `read` operation on property `Holder` of `Account` and ii) a local authorization exported by CDB_2 to perform a `read` operation on property `Balance` of `Accounts`. To ensure consistency, the only global authorization that can be derived on the integrated object `Account` starting from authorizations exported by CDB_1 and CDB_2 is authorization to perform a `read` operation on property `Balance`. In fact, both databases CDB_1 and CDB_2 authorize other federation's users to read accounts' balances, while only CDB_1 authorizes federation's users to read accounts' holders.

2.2 Exported local authorizations

Exported local authorizations specify the privileges federation's users can execute on protection objects of component databases. Each CDB exports a set of local authorizations for its objects, on the basis of local security policies and requirements.

To cope with different authorization paradigms, we consider a local authorization as a triple $auth = \langle s, op, o \rangle$, where s is a subject, op is an operation, and o is an object. In particular, $auth$ states that subject s is authorized to perform operation op on object o.

Subjects in local authorizations can be users, roles, or groups, depending on the authorization paradigm adopted in each CDB_i [Loc88,Fer94]. As a consequence, s can be the identifier of a user ($s = CDB_i.uid$), or the name of a role ($s = CDB_i.n_r$), or the name of a group ($s = CDB_i.n_g$). For subjects, we introduce an operator $USER()$ that returns the set of identifiers of the local users associated with a subject s. In particular, $USER(s)$ coincides with s in case of authorizations specified for single users; $USER(s)$ returns the identifiers of all local users authorized to play the role $CDB_i.n_r$ or to participate in the group $CDB_i.n_g$, in case of authorizations

specified for roles and groups, respectively. In the following, we distinguish between *local subjects*, that is, subjects of some CDB and *global subjects*, that is, subjects defined at the federation level.

Operations in local authorizations can be elementary operations or transactions. We consider the following elementary operations: i) *create*, to create an instance of an object, ii) *read*, to read a structural property of an object, and iii) *write*, to write a structural property of an object. Operations *read* and *write* are defined on structural properties to capture local privileges in a precise way. Note that, in the case of relational model, the *create*, *read*, and *write* operations correspond to the *insert* privilege on a table, and to the *select* and *update* privileges on single columns, respectively.

Transactions are non-elementary operations composed of several elementary operations. Transactions can correspond to SQL queries or to more complex applications on relations and to methods on object classes. For example, with reference to Fig. 2, behavioral property `withdraw` is an example of transaction, which is composed of a `read` and a `write` operation on the `Balance` attribute of a given account. Also for operations, we distinguish between *local operations* and *global operations* to refer to operations on local and global objects, respectively.

Objects in local authorizations are protection objects of a component schema CS_i, that is, local objects / structural properties.

2.3 Structure of the dictionary

To support semi-automatic comparison of local authorizations exported by different CDBs, we maintain a dictionary, where names and semantic relationships between names characterizing federation's elements are stored in a structured way. In particular, the dictionary contains the following sets of names:

- S: the set of subject names;
- O: the set of object names;
- SP: the set of structural property names;
- OP: the set of operation names.

The dictionary is organized in three sections: a *subject*, an *object*, and an *operation section.*

Subject section

In this dictionary section, for each subject name $n_s \in S$, we maintain the set of users associated with the corresponding subject s, i.e., $USER(s)$. For each name $n_s \in S$ denoting a global subject, we maintain the list of the names of its corresponding local subjects, to enforce global authorization propagation, as discussed in Section 5.

Furthermore, in this section of the dictionary, we maintain a (reference to) a predefined thesaurus of role names for the application domain of the federation [Sal89]. Semantic relationships between terms in a thesaurus are used to define new names for global subjects of the federation starting from local subject names (see Section 5). In particular, *synonymy* and *hypernymy* relationships are useful [Bri94].

Two terms are synonyms if they can be interchangeably used as role names in every context, without changes in meaning. For example, `Teller` and `Clerk` are synonyms in the thesaurus. The hypernym of a name n is a name with a broader, more general meaning than n. For example, `Teller` is hypernym of `Teller-in`.

Operation section

Two relationships between operations are relevant for deriving global authorizations: i) equivalence and ii) implication [Rab91].

An *equivalence* relationship between two operations *op* and *op'* indicates that they produce the same effect on different local objects.* An elementary operation is equivalent to itself. Two transactions *op* and *op'* are equivalent if each elementary operation in *op* is equivalent to exactly one elementary operation in *op'*. In case of an object-oriented model, equivalence between methods defined on semantically similar objects can be heuristically evaluated by examining their signature and, possibly, their pre- and post-conditions. In particular, a requirement for equivalence between methods is that their signatures have the same arity and their input and output parameters have the same or synonym names and are defined over the same (or restricted) domains, as discussed in [Rum91,Kim95]. A more precise analysis of equivalence would consider also method's code.

An *implication* relationship between two operations *op* and *op'* indicates that *op* on a given protection object implies *op'* on the same object. As for elementary operations, a `write` operation on a structural property implies a `read` operation on the same property, while a `create` operation on an object implies a `write` operation on all the structural properties of the object. Basic implications between elementary operations in an object-oriented model are discussed in [Rab91]. Implications between transactions can be manually defined by the SA on the basis of their semantics. An heuristic criterion that can be used to determine implications between transactions in different CDBs is based on the analysis of the elementary operations they perform. A transaction *op* implies a transaction *op'* if each elementary operation in *op'* is equivalent to an elementary operation in *op*.

In the dictionary, equivalence and implication between operations are represented by means of a "$\equiv$" relationship and a "$\rightarrow$" relationship between operations names, respectively. In particular:

- For each pair of operations *op* and *op'* that are equivalent, a relationship $\langle n_{op} \equiv n_{op'} \rangle$ is defined between their names $n_{op}, n_{op'} \in OP$ in the dictionary. For instance, with reference to Fig. 2, we define a relationship $\langle CDB_1.\texttt{block} \equiv CDB_2.\texttt{block} \rangle$, since we consider these two operations on accounts equivalent in both office databases.
- For each pair of operations *op* and *op'* such that *op* implies *op'*, a relationship $\langle n_{op} \rightarrow n_{op'} \rangle$ is defined between their names $n_{op}, n_{op'} \in OP$ in the dictionary. For instance, with reference to Fig. 2, we define a relationship $\langle CDB_2.\texttt{release} \rightarrow CDB_1.\texttt{release} \rangle$. In fact, since `release` is composed of a set of `write` operations, one for each structural property of an account, `release` on `Accounts`

*Note that, for our purposes we are interested in the evaluation of equivalence between operations defined on semantically similar objects of different component schemas.

performs a `write` operation more than `release` on `Account`, to write property `Classification`.

Object section

In the object section of the dictionary, names of objects and structural properties are organized according to relationships that represent the information about schema integration. In particular, we define two relationships between federation protection objects: i) similarity and ii) genericity.

The *similarity* relationship between two local objects (structural properties) indicates that they have been integrated into a global object (structural property) in the federated schema.

The *genericity* relationship between a global object (structural property) and a local object (structural property) indicates the correspondence between them as a consequence of the schema integration process.

In the dictionary, the similarity and genericity relationships between protection objects are represented by means of a "SIM" relationship and a "GEN" relationship between protection object names, respectively. In particular:

- For each pair of objects o and o' that are integrated into a corresponding integrated object $\overline{o}$ in the federated schema, a relationship $\langle n_o \text{ SIM } n_{o'} \rangle$ is defined between their names $n_o, n_{o'} \in O$ in the dictionary. For instance, with reference to Fig. 2, we define a relationship $\langle CDB_1.\texttt{Account} \text{ SIM } CDB_2.\texttt{Accounts} \rangle$.
- For each pair of structural properties sp and sp' that are integrated into a corresponding structural property $\overline{sp}$ in the federated schema, a relationship $\langle n_{sp} \text{ SIM } n_{sp'} \rangle$ is defined between their names $n_{sp}, n_{sp'} \in SP$ in the dictionary. For instance, with reference to Fig. 2, we define a relationship $\langle CDB_1.\texttt{Balance} \text{ SIM } CDB_2.\texttt{Balance} \rangle$.
- For each pair of objects $\bar{o}$ and o such that $\bar{o}$ is an integrated object in the federated schema derived from a local object o, a relationship $\langle n_{\bar{o}} \text{ GEN } n_o \rangle$ is defined between their names $n_{\bar{o}}, n_o \in O$. For instance, with reference to Fig. 2, a relationship $\langle FS.\texttt{Account} \text{ GEN } CDB_1.\texttt{Account} \rangle$ is defined.
- For each pair of structural properties $\bar{sp}$ and sp such that $\bar{sp}$ is a property in the federated schema derived from a local structural property sp, a relationship $\langle n_{\bar{sp}} \text{ GEN } n_{sp} \rangle$ is defined between their names $n_{\bar{sp}}, n_{sp} \in SP$. For instance, with reference to Fig. 2, a relationship $\langle FS.\texttt{Balance} \text{ GEN } CDB_1.\texttt{Balance} \rangle$ is defined.

In the dictionary, for each set of names X (X is used to to denote one of the sets S, O, SP, OP), a graph $\langle N_X, E_X \rangle$ is defined, where N_X is a set of nodes and E_X is a set of edges. Nodes represent names of the considered set X, and edges represent relationships among names of X. An edge $e \in E_X$ is defined as a triple $\langle n_i, n_j, l \rangle$ where n_i is the source node, n_j is the destination node, and $l = \Re$ is the label associated with the edge, specifying the type of relationship $\Re$ represented by the edge (i.e., $\equiv$, $\rightarrow$, SIM, GEN). We note that the $\equiv$ and SIM relationships are symmetric and transitive, while the $\rightarrow$ relationship is transitive.

In the following, we will use notation $n_i \Re n_j$ to denote that an edge $\langle n_i, n_j, \Re \rangle$ is defined between n_i and n_j in the dictionary, and notation $n_i \Re^* n_j$ to denote that a

path of length k, with $k \geq 1$, is defined between n_i and n_j in the dictionary for a given relationship $\Re$.

2.4 Global authorization derivation

The approach we propose to global authorization derivation is articulated in the following phases:

1. **Analysis of local authorizations.** In this phase, we group local authorizations of each subject into a *profile.* Subject profiles are defined for all subjects in the involved CDBs, and are used as a logical unit for authorization comparison, to identify, with the help of the dictionary, "compatible" authorizations of different subjects. We discuss this phase in Section 3.
2. **Clustering of subjects.** In this phase, subject profiles are classified on the basis of the compatibility of their local authorizations, using hierarchical clustering techniques. Subject clustering is described in Section 4.
3. **Abstraction of global authorizations.** In this phase, global authorizations are defined starting from clusters of subject profiles. For each pair of profiles in a given cluster, global authorizations are defined by abstracting the local authorizations that are compatible in the considered pair. Federation's subjects for global authorizations are derived from subjects specified in compatible local authorizations. This phase of the methodology is illustrated in Section 5.

3 ANALYSIS OF LOCAL AUTHORIZATIONS

Evaluation of compatibility between local authorizations requires the comparison of their corresponding elements. In particular, we consider the operations and the objects specified in local authorizations. Authorization compatibility is evaluated on the basis of the compatibility of the operations and objects therein contained. Let us now introduce formal definitions of operation and object compatibility.

Definition 1 (Operation compatibility) *Two operations op and op′ are compatible, denoted by* $op \approx op'$*, if and only if one of the following conditions is verified for their names* n_{op} *and* $n_{op'}$ *in the dictionary:*

- $n_{op} = n_{op'}$
- $n_{op} \equiv^* n_{op'}$
- $n_{op} \rightarrow^* n_{op'}$

with $n_{op}, n_{op'} \in OP$.

According to definition 1, two operations are compatible if they are equivalent or if one of them implies the other, directly or indirectly.

Definition 2 (Object compatibility) *Two local objects o and o' are compatible, denoted by $o \approx o'$, if and only if one of the following conditions is verified for their names n_o and $n_{o'}$ in the dictionary:*

- $n_o = n_{o'}$
- n_o SIM* $n_{o'}$

with $n_o, n_{o'} \in O$.

According to definition 2, two objects are compatible if they are the same object or if they are semantically similar objects in different CDBs.

Definition 3 (Authorization compatibility) *Two authorizations, $auth = \langle s, op, o \rangle$ and $auth' = \langle s', op', o' \rangle$, are compatible, denoted by $auth \approx auth'$, if and only if the involved operations and protection objects are compatible, that is,*

$$auth \approx auth' \leftrightarrow (op \approx op') \wedge (o \approx o').$$

For example, let us consider authorizations $auth_1 = \langle CDB_1$.`Teller`, CDB_1.`release`, CDB_1.`Account`$\rangle$ and $auth'_1 = \langle CDB_2$.`Clerk`, CDB_2.`release`, CDB_2.`Accounts`$\rangle$ are compatible because CDB_2.`release` $\rightarrow$ CDB_1.`release` and $\langle CDB_1$.`Account` SIM CDB_2. `Accounts`$\rangle$ in the dictionary.

3.1 Subject profiles

To facilitate the identification of compatible authorizations, for each local subject s, we define a profile $\langle n_s, Auth_s \rangle$, where $Auth_s$ is the set of local authorizations exported for s.

The level of similarity between two subjects in different CDBs is determined on the basis of the number of their authorizations that are compatible in their profiles, as follows.

Definition 4 (Subject similarity coefficient) *The* Subject similarity coefficient *of two local subjects s and s', denoted by $Sim(s, s')$, is the measure of the compatibility between their authorizations, computed as follows.*

$$Sim(s, s') = \frac{2 \cdot \mid \{\langle auth, auth' \rangle \mid auth \in Auth_s, auth' \in Auth_{s'}, auth \approx auth'\} \mid}{\mid Auth_s \mid + \mid Auth_{s'} \mid}$$

where $\mid Auth_s \mid$ denotes the cardinality of $Auth_s$.

According to definition 4, the Similarity coefficient of two local subjects is evaluated using the Dice's function [Sal89] and returns a value in the range $[0, 1]$. In particular, given two subject profiles, the higher the number of authorizations that are compatible, the greater the similarity of corresponding subjects. The Dice's function defines a $1 - 1$ mapping Φ for the authorizations in two profiles

CDB_1.**Teller**
$\{auth_1 = \langle CDB_1.\texttt{Teller}, CDB_1.\texttt{release}, CDB_1.\texttt{Account}\rangle,$
$auth_2 = \langle CDB_1.\texttt{Teller}, CDB_1.\texttt{block}, CDB_1.\texttt{Account}\rangle,$
$auth_3 = \langle CDB_1.\texttt{Teller}, \texttt{read}, CDB_1.\texttt{Holder}\rangle$
$auth_4 = \langle CDB_1.\texttt{Teller}, \texttt{read}, CDB_1.\texttt{Balance}\rangle$
$auth_5 = \langle CDB_1.\texttt{Teller}, \texttt{read}, CDB_1.\texttt{Number}\rangle\}$

CDB_2.**Clerk**
$\{auth'_1 = \langle CDB_2.\texttt{Clerk}, CDB_2.\texttt{release}, CDB_2.\texttt{Accounts}\rangle,$
$auth'_2 = \langle CDB_2.\texttt{Clerk}, \texttt{read}, CDB_2.\texttt{Number}\rangle$
$auth'_3 = \langle CDB_2.\texttt{Clerk}, \texttt{read}, CDB_2.\texttt{Balance}\rangle\}$

Figure 3 Examples of subject profiles in CDB_1 and CDB_2

$\langle n_s, Auth_s\rangle$ and $\langle n_{s'}, Auth_{s'}\rangle$ to be compared. An authorization pair $\langle auth, auth'\rangle$ with $auth \in Auth_s, auth' \in Auth_{s'}$ participates in Φ if and only if $auth \approx auth'$. Φ is total if every $auth \in Auth_s$ is mapped into one $auth' \in Auth_{s'}$ and vice versa. Φ is partial if some authorization remains unmapped in $Auth_s$ or $Auth_{s'}$. The higher the number of pairs $\langle auth, auth'\rangle$ participating in Φ, the greater the value returned by the *Sim* coefficient.

For example, let us consider in Fig. 3, two subject profiles defined for local subjects CDB_1.`Teller` and CDB_2.`Clerk`. The similarity coefficient between CDB_1.`Teller` and CDB_2.`Clerk` is computed as follows:

$$Sim(CDB_1.Teller, CDB_2.Clerk) = \left(\frac{2 \cdot 3}{5+3}\right) = 0.75$$

The pairs of compatible authorizations that participate in the mapping Φ are $\langle auth_1, auth'_1\rangle$, $\langle auth_4, auth'_3\rangle$, and $\langle auth_5, auth'_2\rangle$. Compatibility of $\langle auth_1, auth'_1\rangle$ has been discussed in the previous section. Pairs $\langle auth_4, auth'_3\rangle$, and $\langle auth_5, auth'_2\rangle$ are compatible because they involve the elementary operation `read` on structural properties whose names are related by means of a SIM relationship in the dictionary.

3.2 Semantic correspondences

Semantic correspondences can be established between local subjects of different CDBs on the basis of their similarity coefficients. Given two local subjects s and s', three significant cases can occur:

1. *Semantic equivalence.* Two subjects have semantic equivalence if $Sim(s, s') = 1$. This is the strongest measure of similarity two subjects can have, and indicates that all the examined authorizations of both s and s' are compatible (i.e., a total $1-1$ similarity mapping Φ is defined for authorizations of s and s').

2. *Semantic relationship.* Two subjects have semantic relationship if $0 < Sim(s, s') < 1$. This measure of similarity indicates that subjects perform a number of compatible authorizations. We distinguish *behavioral inclusion* and *behavioral overlapping* between s and s'. Behavioral inclusion, e.g., $s \subset s'$ between s and s' means that the including subject (e.g., s') is authorized for operations compatible to all the ones for which the included subject (e.g., s) is authorized, and for additional operations as well. For example, for roles CDB_1.`Teller` and CDB_2.`Clerk` in Fig. 3 we have CDB_2.`Clerk` $\subset$ CDB_1.`Teller`. In fact, all authorizations of `Clerk` participate in Φ, while authorizations $auth_2$ and $auth_3$ of `Teller` are unmapped. Behavioral overlapping $s \cap s'$ between s and s' means that a subset of compatible authorizations is identified in both s and s'.
3. *Mismatch.* Two subjects are said to be mismatching if $Sim(s, s') = 0$. This is the weakest measure of similarity between subjects, and indicates that s and s' do not have compatible authorizations, and can not considered for global authorization derivation.

4 CLUSTERING OF SUBJECTS

To group local subjects on the basis of their similarity coefficients, we use a hierarchical clustering technique [Eve74]. Hierarchical clustering techniques are usually employed for document classification in information retrieval systems, to facilitate document browsing and retrieval [Sal89]. To apply a hierarchical clustering technique, similarity coefficients for all possible pairs of elements (i.e., local subjects) to be clustered must be computed. For subject clustering, we experimented and selected the single-link technique for its capability of pointing out the existence of compatible authorizations between any pairs of subjects in a given cluster. This is an important aspect to be considered for our purposes, since we are interested in deriving a set of global authorizations as much complete as possible.

The output of the single-link technique is a *similarity tree* of subjects. In the similarity tree, the leaves are the subjects and other nodes identify clusters of similar subjects with an associated numerical similarity value.

According to the single-link clustering technique, the subjects are submitted to pairwise similarity comparisons, and the corresponding Sim coefficients are computed (if we have N subjects profiles, $N \cdot (N-1)/2$ coefficients are computed). All Sim coefficients are stored in a $N \times N$ similarity matrix M; an entry $M[i, j]$ of the similarity matrix M corresponds to the Sim coefficient for two subjects s_i and s_j. The procedure starts by placing each subject in a cluster by its own, giving origin to N clusters. Clusters are iteratively defined, by combining at each iteration the pair of clusters h and k having the greatest Sim coefficient in M. At each iteration, M is properly updated by deleting the rows and columns of clusters h and k and by defining a new row and column $h + k$ for the newly defined cluster. The Sim coefficient in each entry of row (column) $h + k$ is calculated as the maximum value between the Sim values previously stored in corresponding entries of rows (columns) h and k. The procedure terminates when the number of clusters left in M is one.

CDB_2.**Branch-Manager**
$\{auth''_1 = \langle CDB_2$.Branch-Manager, CDB_2.block, CDB_2.Accounts$\rangle$,
$auth''_2 = \langle CDB_2$.Branch-Manager, write, CDB_2.Classification$\rangle$
$auth''_3 = \langle CDB_2$.Branch-Manager, read, CDB_2.Owner$\rangle$
$auth''_4 = \langle CDB_2$.Branch-Manager, write, CDB_2.Interest-rate$\rangle\}$

Figure 4 Subject profile for Branch-Manager in CDB_2

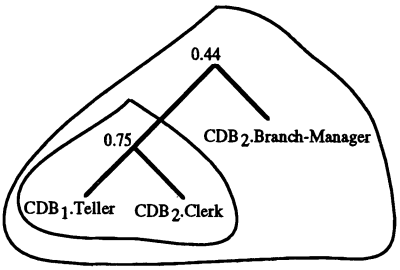

Figure 5 Similarity tree for the subjects of CDB_1 and CDB_2

4.1 Example

To show an example of application of the single-link technique, let us consider, in addition to profiles of Fig. 3, a third subject profile in CDB_2 shown in Fig. 4.

The *Sim* coefficient between all possible pairs of subjects are the following:

$Sim(CDB_1.Teller, CDB_2.Clerk) = 0.75$
$Sim(CDB_1.Teller, CDB_2.Branch - Manager) = 0.\bar{4}$
$Sim(CDB_2.Clerk, CDB_2.Branch - Manager) = 0$

In Fig.5, we present the simple similarity tree obtained by applying the single-link technique to the three subject profiles of Fig. 3 and Fig. 4.

In the similarity tree, subject clusters whose similarity coefficient is different from zero are considered for the derivation of global authorizations. In our case, such clusters are evidenced in Fig. 5.

5 ABSTRACTION OF GLOBAL AUTHORIZATIONS

The term abstraction is used to denote the combination of compatible authorizations into a global authorization. Abstraction is applied to both operations and objects in compatible local authorizations. The following rules are used for authorization abstraction.

Rule 1. Operation abstraction
Let *op* and *op'*, with $op \approx op'$, be two compatible operations. The operation $\overline{op}$ abstraction of *op* and *op'* is defined as follows:

$$\overline{op} = \begin{cases} op' & \text{if } (op \equiv^* op') \vee (op \rightarrow^* op') \\ op & \text{otherwise} \end{cases}$$

According to Rule 1, the operation abstraction of two compatible operations in local authorizations coincides with one of them if they are equivalent, or to the implied operation if one of the two operations implies (directly or indirectly) the other. This rule ensures that the most restrictive operation between two compatible local operations is automatically selected as the global operation.

Rule 2. Object abstraction
Let o and o', with $o \approx o'$, be two compatible protection objects. The object $\bar{o}$ abstraction of o and o' coincides with the integrated object $\bar{o}$ corresponding to o and o' in the federated schema.

The abstraction of global authorizations is performed for each subject cluster having a similarity coefficient different from 0 in the similarity tree. Abstraction rules 1 and 2 are applied to the pairs of compatible authorizations of each pair of subjects in the considered cluster. In particular, for two subjects in a cluster, for each pair of authorizations $auth = \langle s, op, o \rangle$ and $auth' = \langle s', op', o' \rangle$ participating in the mapping Φ, a corresponding global authorization $\langle \bar{s}, \bar{op}, \bar{o} \rangle$ is defined, where:

1. $\bar{s}$ is the global subject. Subjects of global authorizations are roles to provide a flexible authorization paradigm. The choice of names for global roles is based on dictionary thesaurus. In particular, if s and s' are roles/groups, a synonym or a hyperonym of them can be selected from the dictionary as global role name.
2. $\bar{op}$ is the global operation obtained by applying Rule 1 to op and op';
3. $\bar{o}$ is the object obtained by applying Rule 2 to o and o'.

For example, with reference to the similarity tree in Fig. 5, we define two global subjects: FS.`Clerk`, from pair CDB_1.`Teller` and CDB_2.`Clerk`, with the following profile:

FS.**Clerk**
$\{auth_1 = \langle FS$.`Clerk`, CDB_1.`release`, FS.`Account`$\rangle$,
$auth_2 = \langle FS$.`Clerk`, `read`, FS.`Number`$\rangle$
$auth_3 = \langle FS$.`Clerk`, `read`, FS.`Balance`$\rangle\}$

and FS.`Manager`, from the pair CDB_1.`Teller` and CDB_2.`Branch-Manager`, with the following profile:

FS.**Manager**
$auth_1 = \langle FS$.`Manager`, CDB_1.`block`, FS.`Account`$\rangle$,
$auth_2 = \langle FS$.`Manager`, `read`, FS.`Holder`$\rangle\}$

We note that all authorizations of CDB_2.`Clerk` are abstracted into FS.`Clerk` profile, due to the behavioral inclusion CDB_2.`Clerk` $\subset$ CDB_1.`Teller`, while only some authorizations of CDB_1.`Teller` and CDB_1.`Branch-Manager` are abstracted

into FS.`Manager` profile, due to behavioral overlapping CDB_1.`Teller` $\cap$ CDB_1.`Branch-Manager`. Furthermore, global role FS.`Clerk` is authorized to perform operation CDB_1.`release` on the instances of object FS.`Account`, because this transaction is executable on all account instances of both CDB_1 and CDB_2. A global authorization to perform CDB_2.`release` is not automatically extracted for FS.`Clerk` because it would originate an inconsistent authorization state (that is, it would authorize users of CDB_1 to perform more operations than the ones they are authorized for in CDB_1). However, it can be manually specified by the federation security administrator, by negotiating with the involved local security administrators.

In the dictionary subject section, for each defined global subject $\bar{s}$, the following information is specified to enforce access control:

- the set of local subjects (i.e., s and s') from which $\bar{s}$ has been derived, to support authorization mapping during access control;
- the set of local users $USERS(\bar{s})$ that are authorized to play $\bar{s}$. The minimal set of local users authorized to play $\bar{s}$ is the union of the users authorized to play s and s', i.e., $USER(\bar{s}) = USER(s) \cup USER(s')$. Additional federation's users authorized to play a given $\bar{s}$ can be manually specified by the federation security administrator, on the basis of specific security requirements.

The sets of users authorized to play defined global roles are the following: *USER(FS.Clerk)=USER(*CDB_1*.Teller)* $\cup$ *USER(*CDB_2*.Clerk)* and *USER(FS.Manager)=USER(*CDB_1*.Teller)* $\cup$ *USER(*CDB_2*.Branch-Manager).*

5.1 Access control

A global user u that wants to perform a given operation an integrated object in the federated schema, first asks the federation to play a given global role, and then send its access request. An operator $PLAY()$ is defined, that returns, for a given federation's user u, the set of global subjects u is authorized to play, that is, $PLAY(u) = \{\bar{s} \mid u \in USER(\bar{s})\}$. Once u has been authorized to play the requested role $\bar{s}$, his requests are evaluated against the global authorizations specified for $\bar{s}$. For authorized requests, the federated DBMS (FDBMS) is responsible for sending the appropriate local request(s) to the involved CDB(s).

The information on global subjects to be specified in each involved CDB to enforce local access control depends on the authorization autonomy that is adopted in the federation. Different levels of authorization autonomy can be enforced in the federation [Jon94]. With a "full authorization autonomy", we require that a CDB imports the global subjects that are authorized to access its local objects, together with: i) the set of associated users, and ii) the global authorizations defined for $\bar{s}$ properly mapped to local objects, using the dictionary. With a "medium authorization autonomy", no additional information is required at the CDBs. In fact, before sending a local request, the FDBMS maps the global subject into the proper local subject of the involved CDB_i, using the dictionary, and the CDB_i trusts the FDBMS and uses the local subject identity for access control. With "low

authorization autonomy", the FDBMS is the subject of the local access request, and the CDB_i trusts the FDBMS only.

6 CONCLUDING REMARKS

In the paper, we have presented a semi-automatic approach to the analysis and comparison of local authorizations of component databases of a federation, to derive global authorizations for integrated objects of the federated schema. Global authorizations defined in this way are consistent with respect to exported local authorizations. The approach exploits a structured dictionary and a set of proper criteria to evaluate the compatibility of local authorizations referring to different subjects. Clustering techniques are used to facilitate the abstraction of global authorizations and the definition of global roles. Global authorizations defined in this way constitute a basic set of authorizations for the federation, that can be manually refined and possibly extended by the federation security administrator, by negotiating new privileges with the local administrators and/or the owners of the local objects to be protected. Global subjects derived with the proposed approach can constitute the basis for defining new authorizations for local and federated objects of a federated schema, in that they provide a classification of local users.

Supporting tools for the approach are under development, which will be applied and experimented on a set of database schemas of the Italian Public Administration information systems. Future research work to refine the approach will cover the following issues: i) derivation of global authorizations for composite objects; ii) organization of global roles into a hierarchy, starting from derived global authorizations; iii) analysis of local role hierarchies of different databases in the derivation of global authorizations, extending the criteria we proposed for the analysis of security specifications [Cas94].

Acknowledgments
The author wish to thank P. Samarati for her useful suggestions that contribute to improve the quality of the paper.

REFERENCES

[Bat86] Batini,C., Lenzerini, M., and Navathe, S. (1986) A Comprehensive Analysis of Methodologies for Database Schema Integration, *ACM Computing Surveys,* **18**(4).

[Bri94] Bright, M.W., Hurson, A.R., and Pakzad, S. (1994) Automated Resolution of Semantic Heterogeneity in Multidatabases, *ACM Transactions On Database Systems,* **19**(2).

[Cas92] Castano, S. and Samarati, P. (1992) An Object-Oriented Security Model for Office Environments, in *Proc. of the 1992 IEEE Int. Carnahan Conference on Security Technology,* Canada.

[Cas94] Castano, S., Martella, G., and Samarati, P. (1994) A New Approach to Security System Development, in *Proc. of 3rd ACM Workshop on New Security Paradigms,* Little Compton, Rhode Island.

[Cas95] Castano, S., Fugini, M.G., Martella, G., and Samarati, P. (1995) *Database Security*, Addison-Wesley.

[Eve74] Everitt, B. (1974) *Cluster Analysis*, Heinemann Educational Books Ltd, Social Science Research Council.

[Fer94] Fernandez, E.B., Wu, J., and Fernandez, M.H. (1994) User Group Structures in Object-Oriented Database Authorizations, in *Proc. Working Conference on Database Security, IFIP WG 11.3,* J. Biskup, M. Morgenstern, and C. Landewhr (eds.).

[Idr94] Idris, N.B., Qutaishat, M.A., and Gray, W.A. (1994) Integration of Secrecy Features in a Federated Database Environment, in *Database security, VII: status and prospects*, North-Holland.

[Jon93] Jonscher, D. and Dittrich, K.R. (1993) Access Control for Database Federation - A Discussion of the State-of-the-Art, in *Proc. of the DBTA Workshop on Interoperability of Database Systems and Database Applications,* Fribourg, Switzerland.

[Jon94] Jonscher, D. and Dittrich, K.R. (1994) An Approach for Building Secure Database Federations, in *Proc. of the 20th Int. Conference on Very Large Databases (VLDB),* Santiago, Chile.

[Jon95] Jonscher, D. and Dittrich, K.R. (1995) Argos - A Configurable Access Control Subsystem Which Can Propogate Access Rights, in *Proc. of 9th IFIP Working Conference on Database Security, IFIP WG 11.3,* Rensselaerville, New York, USA.

[Kim95] Kim, W., Choi, I., Gala, S., and Scheevel, M. (1995) On Resolving Schematic Heterogeneity in Multidatabase Systems, in *Modern Database Systems-The Object Model, Interoperability and Beyond*, W. Kim (ed.), ACM Press.

[Ham93] Hammer, J. and McLeod, D. (1993) An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous Heterogeneous Database Systems, *Intern. Journal of Intelligent and Cooperative Information Systems*, **2**(1).

[Loc88] Lochovsky, F.H. and Woo, C.C. (1988) Role-Based Security in Database Management Systems, in *Database Security: Status and Prospects,* C. Landwehr (eds.), North-Holland.

[Mor92] Morgenstern, M., Lunt, T.F., Thuraisingham, B., and Spooner, D.L. (1992) Security Issues in Federated Database Systems: Panel Contributions, in *Database security, V: status and prospects*, C.E. Landwehr and S. Jajodia (eds.), North-Holland.

[Rab91] Rabitti, F., Bertino, E., Kim, W., and Woelk, D. (1991) A Model of Authorization for Next-Generation Database Systems, *ACM-Trans. On Database Systems,* **16**(1).

[Rum91] J. Rumbaugh, et al., (1991) *Object-Oriented Modeling and Design,* Prentice-Hall International, Inc..

[Sal89] G. Salton, (1989) *Automatic text processing - The transformation, analysis and retrieval of information by computer,* Addison-Wesley.

[She90] Sheth A.P. and Larson, J.P. (1990) Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, *ACM Computing Surveys,* **22**(3).

[Sie91] Siegel, M. and Madnick, S.E. (1991) A Metadata Approach to Resolving Semantic Conflicts, in *Proc. of the 17th Int. Conference on Very Large Databases (VLDB),* Barcelona.

[Spa92] Spaccapietra, S., Parent, C., and Dupont, Y. (1992) Model Independent Assertions for Integration of Heterogeneous Schemas, *VLDB Journal*, **1**.

[Tem87] Templeton, M., Lund, E., and Ward, P. (1987) Pragmatics of Access Control in Mermaid, *Data Engineering*, **10**(3).

[Wan87] Wang, C.Y., and Spooner, D.L. (1987) Access Control in a Heterogeneous Distributed Database Management System, in *Proc. of 6th IEEE Symposium on Reliability in Distributed Software and Database Sytems*, Williamsburg, VA.

Silvana Castano is an assistant professor of computer science at University of Milano. She received the Ph.D. degree in computer science from Politecnico di Milano, in 1993. Her major areas of research include conceptual models, methodologies, and tools for information system analysis and reengineering, database design, and security system development. She is a member of the IEEE Computer Society. She is a co-author of the book Database Security (Addison Wesley, 1995).