5

Deriving Authorizations from Process Analysis in Legacy Information Systems

Silvana Castano [1] *Maria Grazia Fugini* [2]

[1] *Universitá di Milano*
Dipartimento di Scienze dell'Informazione
Via Comelico 39/41, 20135 Milano, Italy
Email: castano@dsi.unimi.it

[2] *Politecnico di Milano*
Dipartimento di Elettronica e Informazione
P.za Leonardo da Vinci 32, 20133 Milano, Italy
Email: fugini@elet.polimi.it

Abstract

The problem of analyzing security requirements is to be addressed in legacy systems when planned restructuring interventions involve also security aspects. In this paper, we propose a three-level model for authorization analysis and an associated method to extract authorizations from legacy systems. The model allows the security administrator to analyze process authorizations for database accesses at different granularity levels of the involved data. The connection between processes and user roles within organizational units of the legacy system are discussed. The initial results of an experimentation of the approach on a set of processes and databases of the Italian Public Administration information systems are presented.

Keywords

Discretionary access control, Authorization analysis, Legacy information systems.

1 INTRODUCTION

Security of data in distributed and heterogeneous systems, such as Public Administration organizations, has received much attention and has been tackled in the last few years with different focuses, such as database security, communications security, standardization of procedures and devices, individual privacy insurance [ISS95,Jon94]. Public Administration information systems are legacy systems characterized by thousands of co-existing processes and applications, spread among several heterogeneous systems [Aik94]. Ad hoc methods and techniques are required to identify security requirements in legacy systems, with capabilities to take into account also security aspects peculiar of distributed and heterogeneous systems [She90]. In fact, legacy systems can have been developed without security requirements in mind and / or without documenting how security requirements have been implemented. In addition, the personnel with the knowledge required to

understand these systems and how they work may be no longer available, making the identification of security authorizations a crucial activity to be performed with information actually available.

In this paper we present a method for organization-oriented analysis of security in legacy Information Systems of the Public Administration. The analysis aims at making evident existing authorizations of processes on data, in order to verify their consistency with the current organization security requirements.

The method is illustrated basing on the results of a study being conducted in cooperation with the Italian National Consortium for Informatics (CINI) and the Italian National Research Council. The study is aimed at devising methods and tools for evaluating existing security measures, and possibly developing new measures, in Information Systems of some key organizations in the Italian Public Administration.

In particular, the study is being performed with the Labour and the Justice Ministries also through the coordination of the "Information Systems Authority for Public Administration" (AIPA). Starting from a large set of data made available by AIPA, we have analyzed *processes*, *Organizational units*, and *databases* belonging to the Labour Ministry. The legacy Information Systems thus considered allowed us to perform an analysis of security of *business procedures*, and of *data* stored in the Ministry databases. The purpose of the study is to analyze the *security requirements* in a Public Administration Information System and to propose a business security model able to fulfill a twofold objective:

- to express the authorizations of the analyzed legacy systems, thus allowing security designers to match them against security requirements and possibly modify some authorizations;
- to be a reference model for the Public Administration in the development of security of its systems.

In this paper, we present the results of the first part of this study, regarding the analysis of authorizations of Public Administration processes on databases. The organizational units are identified; their analysis using a role-based model is a subsequent phase of the project and, hence, is not discussed here. Issues of data distribution and database federation have been studied in a preliminary approach in [Cas96]; they are planned to be integrated within the project together with role and organizational unit analysis.

The paper is organized as follows. In Section 2, we describe the application context of our approach. In Section 3, we illustrate the analysis method adopted for identifying process authorizations in our legacy systems. In Section 4, we describe possible uses of our method in the framework of the Public Administration domain. Finally, in Section 5, we give our concluding remarks.

2 APPLICATION CONTEXT

The elements characterizing our application context are described by means of an ER schema, shown in Fig. 1 [Bat96]. In particular, in the schema we identify:

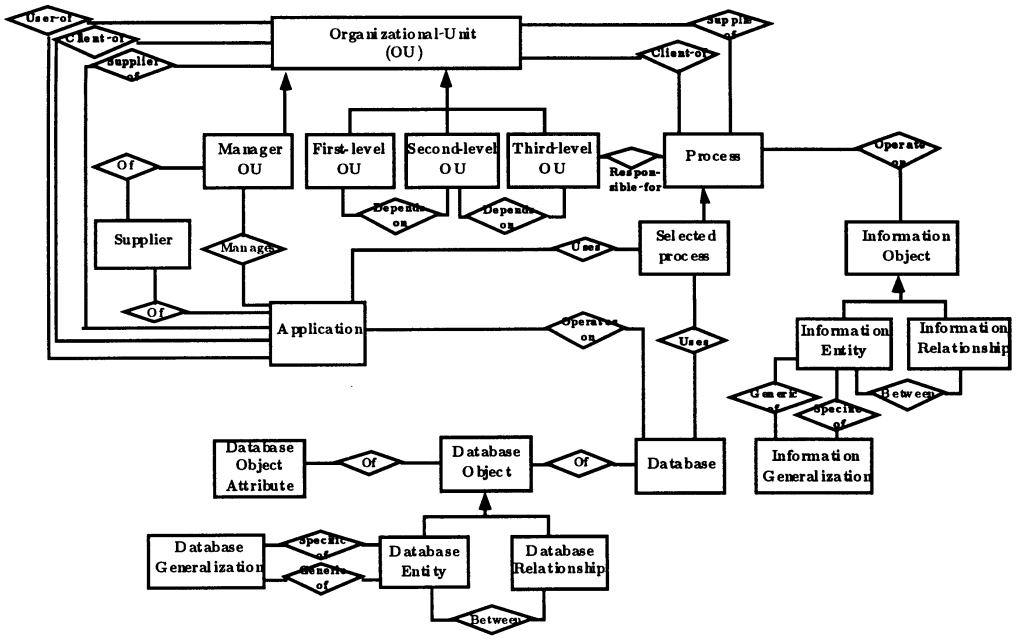

Figure 1 Model of the organizational units and related processes

- *Organizational units*, that is, the users of the information systems. Three levels of organizational units are distinguished, namely Ministries, Divisions, Offices to reflect their hierarchical/functional placement in the organization chart of the Public Administration (PA).
- *Processes*, that is, groups of activities (partly manually executed, partly computer supported) performed to provide services to internal and/or external users of the PA. Organizational units can be responsible for, clients or suppliers of work processes. Processes have an associated ER schema describing the information manipulated by the process in terms of entities and relationships between entities.
- *Applications*, which are computer based and are managed by the EDP manager organizational units; they have client, supplier and user organizational units associated with them.
- *Information objects* exchanged and manipulated by processes, distinguishing between paper based information objects and computer based information objects, all together represented in terms of ER conceptual schemas.
- *Databases*, which are used by the applications. An ER schema is defined for each database describing database structure at the conceptual level.
- *Database objects*, corresponding to the computer based information objects manipulated by the processes.

The data selected for the analysis of security requirements are provided by the Ministry of Labour through AIPA and consist of:

- the description of organizational units of the Ministry of Labour;

- the description of about 100 application processes executed by these organizational units on data;
- the description of 9 databases given as schemas and subschemas according to the Entity-Relationship (ER) model [Chen76].

Our analysis on these descriptions has the purpose of identifying which organizational units can execute which processes and therefore can access which data items; this analysis is performed by coupling the descriptions of the ER process schemas with the ER database schemas. Then, by identifying which organizational unit executes which processes, the aim is to *derive* the existing authorizations between processes and data. These authorizations will be expressed as a triplet $\langle s, op, o \rangle$ where s is a subject, op is a type of access or operation, and o is an object. Initially, the *type of access* is expressed in terms of *transactions* on database schemas and subschemas; subsequent refinements lead to identify process authorizations in terms of basic privileges (read, write, create) on schema elements and eventually on data items.

For the analysis, in the following section we illustrate the three-level authorization model and the associated methodology to identify the existing access modes from processes onto databases.

3 ANALYSIS MODEL AND METHODOLOGY

The three-level authorization model is depicted in Fig. 2 (adapted from the ER security model proposed in [Oh95]). A special type of relationship, called *security relationship* (shown in grey in the figure) is introduced to represent at the conceptual level the privileges that can be executed by a subject on a given object. In our context, privileges are associated with processes. In fact, since we deal with legacy systems, it is very difficult to specify data access privileges directly for user roles, because this would require an in-depth analysis of work procedures in each involved PA office. In legacy systems, role privileges on data can be derived from the authorizations of processes that roles are authorized to execute. Roles which, in an organizational unit, are authorized to execute one or more processes (*process authorizations* in Fig. 2) will, consequently, acquire data access privileges associated with these processes. Issues related the definition of process authorizations are not discussed here, since this task will be performed in a subsequent phase of the project, on the basis of the role hierarchies defined within every single organizational unit.

A *refinement* method is defined for the analysis, leading to show existing authorizations of processes on single data items in terms of elementary access operations (read, write, create). A refinement method is necessary since, dealing with legacy systems, security authorizations are not explicitly stated, but rather, are implicit in the system workflow. Consequently, identification of authorizations for single data items can only be derived starting from a higher level analysis of process functionality and required accesses to existing databases. According to this refinement method, first we identify authorizations for processes to access database schemas (Fig. 2(a)), called *database authorizations*. Then we refine database authorizations

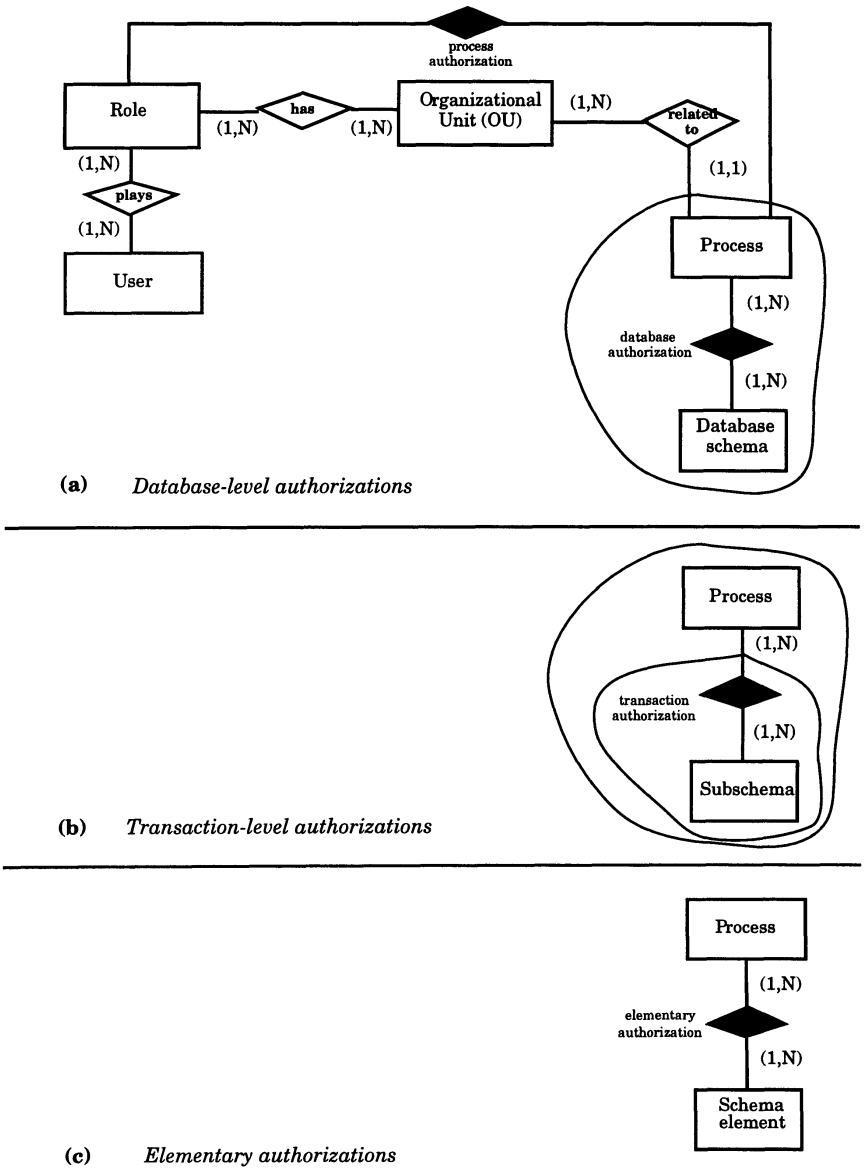

Figure 2 Three-level authorization model

into authorizations for processes to perform operations on subschemas (Fig. 2(b)), called *transaction authorizations.* Eventually we obtain, through a further refinement, authorizations for processes to perform elementary operations on schema elements (Fig. 2(c)), called *elementary authorizations.*

The methodology we propose for the analysis of security requirements is articulated in the following phases:

1. *Identification of database authorizations.*
   In this phase, we identify which databases are accessed by which processes and

organizational units. The analysis is performed by exploiting the ER schemas associated with processes and databases. We describe this phase in Section 3.1.

2. *Identification of transaction authorizations.*
   In this phase, we analyze the functionality of processes in order to identify which *transactions* are executed by a given process on the involved databases. The analysis is performed by exploiting a textual description of process functionality and the involved database schemas. We describe this phase in Section 3.2.
3. *Identification of elementary authorizations.*
   In this phase, for each process transaction identified in the previous phase, we derive the elementary operations involved in the execution of the transaction. The analysis is performed by exploiting the query associated with a transaction and the corresponding database schema(s). We describe this phase in Section 3.3.

3.1 Identification of database authorizations

The goal of this phase is to identify: i) which databases can be accessed by each process of a given OU, and ii) which databases can be accessed by each OU. To this end, we analyze ER schemas associated with processes and ER database schemas to find a match between them. The analysis is performed separately for each OU and, within a given OU, for all processes pertaining to it. In particular, for a given process P_i, starting from elements (e.g., entity, relationship) specified in its corresponding schema, we analyze database schemas to select those containing elements matching P_i's schema elements. A process P_i can access a single database or several databases, depending also on the type of process, namely elementary process or macroprocess. Elementary processes perform an elementary task, with a well defined objective. Macroprocesses perform complex activities whose objective is pursued by means of the coordinated execution of a set of constituent (elementary) processes.

As the result of analyzing ER schemas, we identify a set of *database authorizations*, $DBAUTH = \{\langle s, op, o\rangle\}$, where:

- s can be a process P_i or an organizational unit OU_j;
- op is *access-DB*;
- o is a database DB_k.

A database authorization $\langle P_i, access - DB, DB_k\rangle$ specifies that process P_i is authorized to access database DB_k, because it performs at least one operation on data stored in DB_k.

For example, let us consider the process `Statistics Elaboration for Employment Analysis` (P_1) which belongs to the organization unit `DIII` (OU_1) of the Ministry of Labour and is responsible for producing statistics regarding companies and related employees. Process P_1 accesses the `MC` database (DB_1) whose schema is shown in Fig. 3, to retrieve necessary company and employee data. As a consequence, we can derive the database authorization $\langle P_1, access - DB, DB_1\rangle$ shown in Fig. 3, according to the authorization model illustrated in Fig. 2.

Authorizations of the form $\langle OU_j, access - DB, DB_k\rangle$ can be derived specifying that organization unit OU_j is authorized to access database DB_k. An authorization $\langle OU_j, access - DB, DB_k\rangle$ can be derived in the set $DBAUTH$ only if at least one

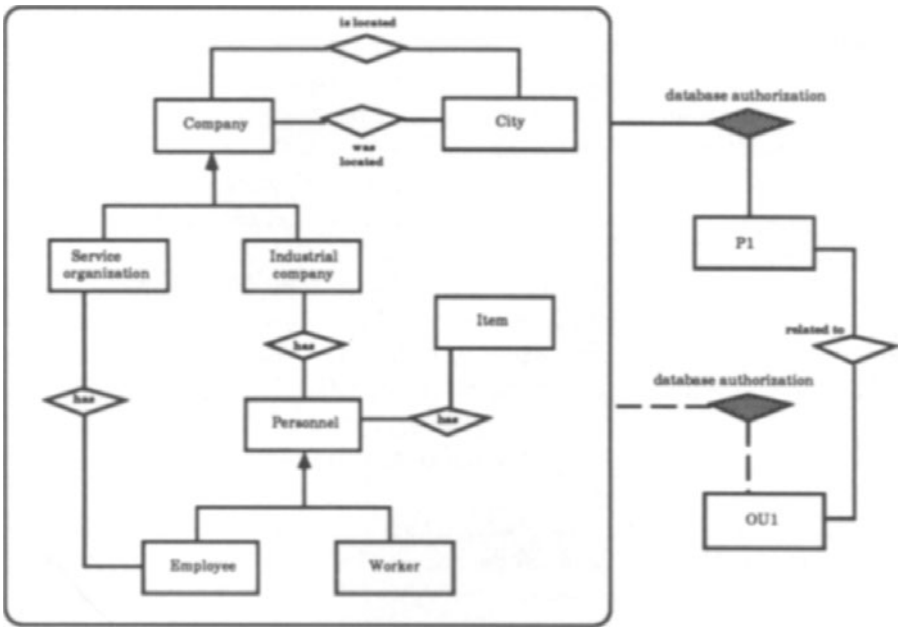

Figure 3 Example of database authorizations

authorization $\langle P_i, access - DB, DB_k \rangle$ is defined in the set $DBAUTH$, where P_i is a process related to OU_j. Database authorizations for OUs are derived in order to perform the security organizational analysis which will be one of the goals of future work. In Fig. 3, the derived authorization $\langle OU_1, access - DB, DB_1 \rangle$ is shown with dashed lines.

3.2 Identification of transaction authorizations

In this phase, we are interested in identifying the groups of operations (transactions) performed by a given process P_i on the database subschemas associated with P_i in database authorizations. For this purpose, the process functionality must be analyzed. For such analysis, a textual specification of process functionality is available in our project, describing the main characteristics of process activity. Referring to available data, we manually identify the main operations performed by the process on a corresponding database(s) by isolating relevant information. In particular, verbs and names of database elements are isolated, denoting the type of operation and the involved data. For each verb and associated database elements, we define a SQL query on the corresponding database subschemas.

One or more SQL queries can be defined for each process P_i, depending on the complexity of the activity performed by P_i. Each defined query corresponds to a *transaction.*

As an example, let us consider process P_1 performing a transaction T_{11} `Company size analysis` to produce an aggregated report giving the number of the industrial companies grouped by size (small, medium, large). The SQL query corresponding to $T_{11}$is the following:

T_{11}: `SELECT COUNT(*)`
`FROM` Company, Industrial Company
`WHERE` Company.Code=Industrial Company.Code
`GROUP BY` Company.Size

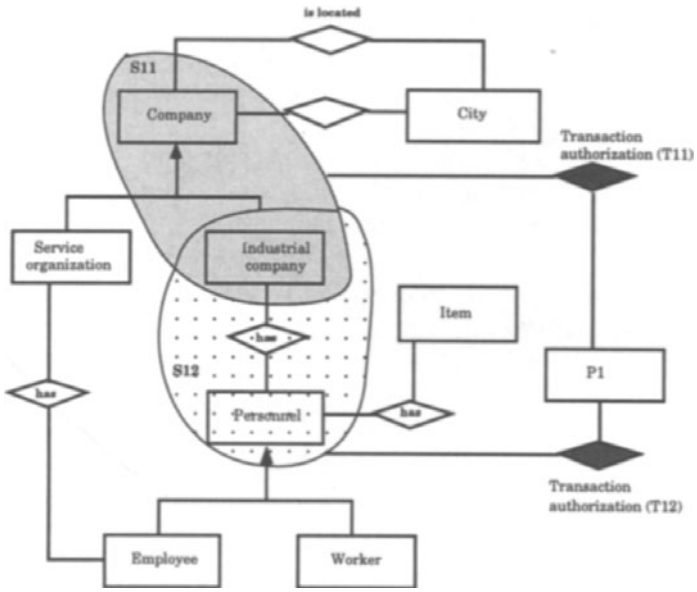

Figure 4 Example of transaction authorizations

As a result, we define a set of *transaction authorizations*, $TAUTH = \{\langle s, op, o\rangle\}$, which is a refinement of the set $DBAUTH$ of database authorizations defined in the previous phase. In particular, for each database authorization $\langle P_i, access - DB, DB_k\rangle \in DBAUTH$ we define one or more transaction authorizations $\langle s, op, o\rangle \in TAUTH$ where:

- s is the process P_i;
- op is a transaction T_{iq} executed by P_i, with $q = 1, \ldots, n$, being n the total number of transactions performed by P_i;
- o is a subschema S_{kt} of the schema S_k associated with database DB_k. S_{kt} contains element(s) of S_k specified in the SQL query associated with T_{iq}.

A transaction authorization $\langle P_i, T_{iq}, S_{kt}\rangle$ specifies that process P_i is authorized to perform operation T_{iq} on (sub)schema S_{kt} of database DB_k.

For example, with reference to process P_1, we define the following transaction authorizations: $\langle P_1, T_{11}, S_{11}\rangle$, where S_{11} denotes the subschema of DB_1 on which T_{11} operates (evidenced by the grey area in Fig. 4), and $\langle P_1, T_{12}, S_{12}\rangle$ where T_{12} is another transaction, named `Personnel/Industrial Company analysis` operating on the subschema S_{12} (dotted area of Fig. 4 according to the following SQL query:

T_{12}: SELECT COUNT(*)
FROM Personnel, Industrial Company
GROUP BY Industrial Company.Code

3.3 Identification of elementary authorizations

In this phase, we further refine transaction authorizations to identify elementary operations performed by a given process P_i on database elements during each transaction T_{iq}. We consider the following elementary operations:

- *create*, to create an instance of an element in the database,
- *read*, to read an (attribute of an) element, and
- *write*, to write an (attribute of an) element.

The *read* and *write* privileges are defined to the level of ER attributes. The *create*, *read*, and *write* operations correspond to the *insert* privilege on a relational database table, and to the *select* and *update* privileges on single table columns, respectively.

For a process P_i, we analyze each of its associated transactions together with the corresponding database schema. For each transaction T_{iq}, elementary operations performed by T_{iq} on each schema element are identified by exploiting the SQL query for T_{iq}.

As the result of transaction analysis, we define a set of *elementary authorizations*, $EAUTH = \{\langle s, op, o\rangle\}$, which is a refinement of set $TAUTH$ of transaction authorizations defined in the previous phase. In particular, for each transaction authorization $\langle P_i, T_{iq}, S_{kt}\rangle \in TAUTH$ we define one or more elementary authorizations $\langle s, op, o\rangle \in TAUTH$ where:

- s is the process P_i;
- op is an elementary operation, that is, $op \in \{create, read, write\}$;
- o is an element or an element attribute of S_{kt}.

An elementary authorization $\langle P_i, op, o\rangle$ specifies that process P_i is authorized to perform the elementary operation op on the corresponding schema element o of database DB_k.

With reference to transaction T_{11} of process P_1 previously specified, we define the following elementary authorizations (see Fig. 5):

$\langle P_1$,*read, Company.Code*$\rangle$
$\langle P_1$,*read, Industrial Company.Code*$\rangle$
$\langle P_1$,*read, Company.Size*$\rangle$

For the sake of simplicity, in Fig. 5, we show elementary authorizations on entities rather than on entity attributes.

The notion of authorization implication [Rab91] is now adopted to relate elementary authorizations. Let e_p be an element (i.e., an entity or a relationship) of

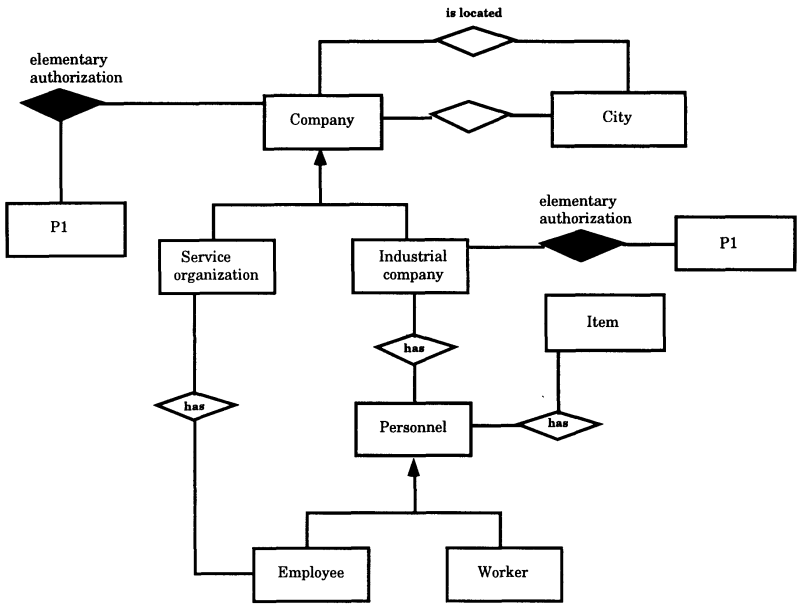

Figure 5 Example of elementary authorizations

a (sub)schema, and $e_p.a_i$ an attribute of e_p. We define the following authorization implications (denoted by symbol "$\rightarrow$"):

- $\langle s, op, e_p \rangle \rightarrow \langle s, op, e_p.a_i \rangle$, with $op \in \{read, write\}$ and $i = 1, \ldots, N$, being N the total number of attributes of the considered element e_p.
- $\langle s, create, e_p \rangle \rightarrow \langle s, write, e_p.a_i \rangle$, for a subset or all attributes of the considered element e_p.

Authorization implication allows one to simplify both the specification and the analysis of authorizations by reducing the number of authorizations of a database. Further implications are under study, together with algorithms for their use.

4 APPLICATIONS OF THE APPROACH

In this section, we discuss the main applications of the analysis methodology presented above. In particular, the methodology and the associated authorization sets can be employed to:

- Derive the existing authorizations of processes on data at different granularity levels, from database level to data item level. This allows the security administrator to identify the security-relevant data items in existing databases.
- Match derived authorizations against the security requirements and policies of the organization. In particular, by aggregating authorizations *by process* the method can show whether a given policy about data access and administration is fulfilled or not. For example, if the minimum-privilege policy [Cas95] must be enforced, the analysis can highlight the data accessed by each process. The security

administrator can then evaluate the adequacy of reported authorizations against the ones required by the policy. Furthermore, by aggregating process authorizations *by organizational unit*, the administrator can check the policy enforcement considering also user roles.

- Support the analysis of role privileges. In order to be effective, the *process-oriented* analysis methodology should be coupled with an *organization-oriented* security analysis in terms of roles and organizational units [Hol95,ISS95]. To this end, Role-Based Access Control models (RBAC) are generally used [San96]. In these models, a role is a job function within an organization describing the authorization conferred to users and is, therefore, suitable to focus the organization structure and its connection to permissions. Moreover, RBAC are a flexible and application-independent paradigm able to accommodate various policies and different applications with minimal customization; it seems therefore a good candidate to become a reference framework for security. RBAC can then be interpreted either into a mandatory or a discretionary access control, depending on the particular organization and on existing mechanisms. Finally, RBAC are based on graphs and hierarchies enabling one to analyze the existing and desired permissions. RBAC is available in some commercial DBMSs and is therefore accessible for experimentation.
  The main advantages of RBAC for our purposes is the ability to represent the roles of users in organizational units, according to the existing organization chart. This allows the security administrator(s) to study the user hierarchies and their actions upon data, through authorization to execute processes.

5 CONCLUDING REMARKS

In this paper, we have presented a methodology for deriving authorizations to access data at different granularity levels on the basis of process analysis. The method described in the paper is intended to be an analysis tool for deriving existing authorizations in legacy information systems and for verifying their adequacy to organization security policies. The method has been illustrated with reference to our experience with some Italian Public Administration information systems.

An environment to support the analysis method previously illustrated has been developed, based on a repository developed by AIPA for storage of ER schemas and other data associated with processes in different organization units. The implementation environment is PC-based, using Access 7.0. The AIPA repository provides functionalities for visualization of process and database schemas and associated information, both with textual information format and with a simple graphical editor for ER schemas. Our analysis method has been experimented on a sample of 30 process specifications related to the Labour Ministry. On top of the AIPA repository, a *toolkit* of SQL queries has been developed to support the three-level based authorization analysis.

Future research work will be devoted to the analysis of user roles in organizational units to identify connections between roles and processes. Experimentation using commercially available DBMS packages will be performed. In addition, security requirements and policies will be collected through interviews to selected Public

Administration offices in order to match them against the authorizations derived by our method. A further issues to be investigated regards the applicability of a finer analysis of process functionality, based on workflow modeling techniques to consider also aspects of data distribution and heterogeneity [Geo95].

Acknowledgments
This work has been partially supported by the Italian Consortium for Informatics (CINI) and by the Italian National Research Council in the framework of "Progetto Strategico Informatica nella Pubblica Amministrazione - DEMOSTENE Project". We thank doctoral students who contributed to test the method and to implement support tools.

REFERENCES

[Aik94] Aiken, P., Muntz, A., and Richards, R.. (1994) DoD Legacy Systems - Reverse Engineering Data Requirements. *Communications of the ACM,* **37**(5).

[Bat96] Batini, C., Castano, S., De Antonellis, V., Fugini, M.G., and Pernici, B. (1996) Analysis of an Inventory of Information Systems in the Public Administration. *Requirements Engineering Journal,* **1**(1).

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

[Cas96] Castano, S. (1996) An Approach to Deriving Global Authorizations in Federated Database Systems. In *Proc. of 10th Annual IFIP WG 11.3 Working Conference on Database Security*, Como, Italy.

[Chen76] Chen, P.P. (1976) The Entity-Relationship Model: Towards a Unified View of Data. *ACM Trans. on Database Systems,* **1**(1).

[Geo95] Georgakopoulos, G., Hornik, M., and Sheth, A. (1995) An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases,* **3**.

[Hol95] Holbein, R., Teufel, S., and Bauknecht, K. (1995) The Use of Business Process Models for Security Designs in Organisations, in [ISS95].

[ISS95] (1995) *Information Systems Security - Facing the information society of the 21st Century*, Proc. of IFIP /SEC'95, 12th Int. Information Security Conference, S.K. (Eds. Katsikas S.K. and Gritzalis, D.), Chapman&Hall.

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

[Oh95] Oh, Y.C., and Navathe, S.B. (1995) SEER: Security Enhanced Entity-Relationship Model for Secure Relational Databases. In *Proc. of OO-ER'95, Int. Conf. on the Object-Oriented and Entity-Relationship Modelling,* LNCS n.1021, Gold Coast, Australia.

[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).

[San96] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., and Youman, C.E. (1996) Role-Based Access Control Models. *IEEE Computer*, February.

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