A Formal Model for Inter-Organizational Data Coherence Maintenance *

F.Arcieri[1], E.Cappadozzi[2], G.Melideo[3],
P.Naggar[2], E.Nardelli[3,4], and M.Talamo[2,5]

[1] Consultant to AIPA for the SICC
("Sistema di Interscambio Catasto-Comuni") project
[2] "Coordinamento dei Progetti Intersettoriali" of AIPA - "Autorità per l'Informatica nella Pubblica Amministrazione", Roma, Italy
[3] Dipartimento di Informatica, Univ. of L'Aquila, L'Aquila, Italy.
{last-name}@univaq.it
[4] Istituto di Analisi dei Sistemi ed Informatica, C.N.R., Roma, Italy.
[5] Dipartimento di Matematica, Univ. of Roma 2 "Tor Vergata", Roma, Italy.
talamo@mat.uniroma2.it

Abstract. In this paper we present a formal model supporting cooperation between information systems of autonomous and independent organizations. One of the major issues in this area is how to ensure the coherence in the overall (distributed) set of data. In fact, these data items are clearly overlapping in many cases, since organizations share a same portion of the reality of interest, but their representations are autonomously and independently managed by each organization. The general architectural framework introduced in a previous paper [2] for this purpose is here formalized and extended to support both detection of and recovery from data incoherence. Our solution, which allows to follow an incremental route to coherence enforcement, is rooted in experiences of development of inter-organizational cooperative information systems managed by the "Coordinamento dei Progetti Intersettoriali" of AIPA, the Italian Authority for Information Technology in Public Administration.

Keywords: Data Coherence Maintenance, Data Interoperability, Federated Databases, Information Systems Integration, Inter-organizational Cooperation.

1 Introduction

Interoperability has been a subject of database research since a long time, with a variable level of attention. In the last years, given the unavoidable issue of legacy

* Paolo Naggar is currently with Gruppo CM, Roma, Italia, where he is the Director of the Innovation Division. Maurizio Talamo has been the CEO of the "Coordinamento dei Progetti Intersettoriali" of AIPA during the development of the research here described.

systems and the explosion of network connectivity, its relevance has again increased (e.g., see the special issue [20]). The whole area of information systems integration is a highly researched topic [14], with the goal of allowing users to make an effective use of the wealth of data easily reachable through communication networks.

A recent series of workshops on engineering federated database systems and information systems [11–13] has pointed out that one of the open research issues in this area is the integration of legacy databases in a federation of autonomous heterogeneous information systems and the ability of maintaining coherence of information representation during evolution over time of the systems involved in the federation.

The interoperability scenario we make reference to is the following: (i) there are many large and autonomous organizations which have to cooperate so that each one of them is able to reach its own goals, that are usually set by some external entity (e.g., a law, a statute); (ii) an organization is not the source/manager of all the data it needs to reach its own goals: interaction with other organizations to obtain data that are needed always happens through the exchange of certified messages; in some cases the interaction flow is specified by a law; (iii) there is no real possibility of forcing and/or coordinating changes in the legacy information systems of the various organizations.

In this scenario, the most critical issue is how to enforce and maintain the *coherence of the overall (distributed) set of data*. In fact, each organization will develop and maintain internally databases for information needed for its activities, without having a full control of its evolution. This situation will produce incoherence in the overall set of data, sooner or later, with absolute certainty.

The above described scenario is typical of the Public Administration of many western world countries. The large organizations are federal and national agencies. An example of a goal of an organization is to support a government action (e.g., to help government in deciding about changes in taxation rates for the various income levels) or to provide a specific service to citizens (e.g., to keep a citizen up-to-date with its rights and duties regarding his/her tax declaration). The scenario might apply also to newer organizational structures (e.g., the network enterprise) for very large companies.

Different organizations may have different representations of the same (part of) universe of discourse. Then the issue is how to keep the correlation and synchronization among representation items that, in the various organizations, refer to the same element of the reality of interest but are subject to independent evolution dynamics.

Since the lack of coherence derives mainly from the organizational framework, then the technical solution has to be designed in a way to match needs and behaviour of the organizations involved. Moreover, the technical solution for coherence maintenance has to be designed so that the overall system has good performances and both technical and organizational costs of cooperation are not hidden.

In terms of data interoperability this distributed coherence problem lays somewhere between the syntactic and the semantic levels. This means that we assume that syntactic homogeneity has been or can be reached, using wrappers [18] if needed. Our goal, then, is not to reach semantic homogeneity (see the special issue [16] for a recent discussion of semantic interoperability issues and approaches, or [10] for information integration at the conceptual level) but just to keep the representations of the same data items of the reality of interests aligned in the various databases of the cooperating organizations. Semantics of data is important to reach our goal, but it is always taken into account through direct human intervention.

It is also important to stress that we are not dealing with a view maintenance problem. A view, in fact, in the standard interpretation of the term, allows various users to see the same set of data from different viewpoints [1]. Under this respect our case is different since, even if various Source Databases share a common semantic background they focus, in general, on largely different sets of data since they are within completely different organizations. Consider, for example, a cadastral database (having a focus on land parcels and properties) and a people's personal data database (having a focus on people and families).

In this paper we formalize and extend the general framework introduced in [2] dealing with these issues. Moreover, we briefly present the dynamic aspects, that is the means by which incoherence can be recovered once it is detected, which are the focus of a recent paper [8]. Our solution is rooted in experiences of development of inter-organizational cooperative information systems projects managed by the "Coordinamento dei Progetti Intersettoriali" of AIPA, the Italian Authority for Information Technology in Public Administration. Systems SICC [3, 19] – for cadastral data exchange among italian Municipalities, Ministry of Finance, Notaries and Certified Land Surveyors, SCT [4, 5] – for territorial data exchange among Public Administrations, and SIM [6] – providing e-government services to people living in mountain areas, have all been designed and implemented according to this approach.

We stress the fact that our approach makes it possible to evaluate and tune the impact on performances of a given Source Database deriving from cooperation-triggered requests. In general, implementing interoperability among Source Databases so to provide acceptable performance is a highly challenging task [15], since usually access number and access policies required to an underlying Source Database by the execution of coherence maintenance functions are largely out of the control of the designer. On the contrary, with our model it possible to perform a rightsizing of the overall system through a cost-benefit analysis.

2 The formal model

We assume, from now on, that the cooperating organizations have their data stored in relational databases. This is not always the case, above at all for older legacy information systems. But we can always assume that difficulties of dealing

with non-relational technology are tackled by using *wrappers* [18] to encapsulate underlying data, to hide the physical details of how source database have been designed and implemented, and to expose data for cooperation as if they come from relational tables.

Let U be the whole reality of interest for the set of databases one wants to make interoperable, called also the *universe of the discourse* or the *reality of interest*. Any element in U has associated values for some *features* (in general, a very large number of them). Examples of features are the given name of a person, the color of a car, the owner of a book, and so on. Relations in the various Source Databases represent (elements of) U by storing values for various features of the elements of U as values of their attributes. Each Source Database of course represents the features that are more relevant to the modeled fragment of the reality of interest. Since, by assumption, all the Source Databases share U, the same feature value of a specific object may be represented many times. The coherence issue is to ensure that all these various representations in various Source Databases are aligned.

For the purposes of interaction between various Source Databases we call *Supplier* any attribute generating the value of a feature or entitled to change it, while *User* is any attribute interested only in using such a value.

In order to group attributes referring to the those features characterizing a same element of U a label called *role* is attached to each attribute: the role therefore specifies how attributes are aggregated in the same relation to identify, through their values, an object of U.

Our approach defines a framework allowing to relate values of the same feature for the same element of U, in the various Source Databases, with respect to the considered set of Suppliers, so that it becomes possible to introduce mechanisms for incoherence detection among them and between them and the Users.

2.1 Specification

The formal model is based on the following conceptualization. Let R be the set of relation schemes in the Source Databases we consider for interoperability, and let A be the set of attribute names in R[1]. We assume the domain of values for A is a universe D representable in a logical data model for databases[2]. We admit null values as possible values, but we do not distinguish among their various possible flavours, and simply assume value null belongs to D.

An *Access Keys Scheme* (AKS) over the couple (R, A), denoted $\Sigma(R, A)$ (or simply Σ when no ambiguity arises) is defined by its *signature* and has an *interpretation*.

[1] Just for the sake of clarity of presentation we assume all attribute names are distinct and all relation schemes have distinct names: but these assumptions can be omitted and all results are still valid.

[2] We assume that the extension of D is not changing during the time-life of all the databases we consider for interoperability: this simply means that at no point in time new and unknown values will be found in any Source Database. With a proper definition of D this goal can always be met.

Signature. The signature of $\Sigma(R, A)$ is a quintuple $\langle \Phi, \mathrm{P}, \mathcal{F}, \mathcal{R}, S, \rangle$ where:

- Φ is a finite set of *feature* names;
- P is a finite set of *role* names, $\Phi \cap \mathrm{P} = \emptyset$;
- $\mathcal{F} : A \mapsto \Phi$ is the function providing for each attribute a in A the unique feature name $\mathcal{F}(a)$ associated to it;
- $\mathcal{R} : A \mapsto \mathrm{P}$ is the function providing for each attribute a in A the unique role name $\mathcal{R}(a)$ associated to it;
- $S \subseteq A$, denotes the set of *suppliers* for attribute values, $A \setminus S$ is called the set of *users*.

In the rest of the paper, for an attribute $a \in A$ we write ϕ_a to denote the feature name associated with a (i.e., $\mathcal{F}(a) = \phi_a$, $\phi_a \in \Phi$), ρ_a to denote the role name associate with a (i.e., $\mathcal{R}(a) = \rho_a$, $\rho_a \in \mathrm{P}$) and $r_a \in R$ to denote the relation containing a (written also as $a \in r_a$). For a relation $r \in R$ we write $\mathrm{ext}(r)$ to denote the set of tuples belonging to the extension of r and we let $\mathrm{ext}(X) = \cup_{r \in X} \mathrm{ext}(r)$, for each $X \subseteq R$. Given $a \in r$ and $t \in \mathrm{ext}(r)$, we denote with $t.a \in D$ the value taken by tuple t in correspondence with a. For shortness, given a role ρ_a we denote with r_{ρ_a} the relation r_a the role is referring to indirectly through attribute a. For every $B \subseteq A$, B_ϕ denotes the set of attributes of B referring to the same feature ϕ, i.e., $B_\phi = \{a \mid \phi_a = \phi, a \in B\}$. As a particular case, S_ϕ denotes the set of suppliers referring to ϕ. Finally, given a feature ϕ, we let $R_\phi = \{r_a \mid a \in A_\phi\}$.

There are the following *soundness* conditions for $\Sigma(R, A)$: (i) $\forall a, b \in A$ it is $\rho_a = \rho_b \Rightarrow r_a = r_b$, that is attributes sharing the same role have to belong to the same relation, and (ii) $\forall \phi \in \Phi$ it is $\exists a \in S$ such that $\phi_a = \phi$, that is there has to be a supplier for each feature.

Example 1. We use as a running example three Source Databases with the following involved tables: Properties(O, M, P-A, P-N, C) in the Ministry of Finance's DBMS, Apartments(P, A-A, A-N, A) and LocalTax(L-A, L-N, T) in a Municipality's DBMS. See in figure 1-(a) an instance of these tables representing an apartment of 180 square meters in New York, located in Main St. 1, with apartment number 14, owned by Mr. Brown and paying taxes according to fiscal category A2. In our scenario, it may happen that after an update in the Ministry of Finance's DBMS (the apartment is divided in two smaller ones) the situation becomes the one depicted in figure 1-(b) and it is clear that it is no more possible now to automatically match the two new tuples in Properties with the old one in Apartments.

For a more complete discussion of this example and the dynamics of events leading to incoherence generation see [2, 3].

In this example we have for feature names: ϕ_{O} = 'names of people' and $\phi_{\mathsf{O}} = \phi_{\mathsf{P}}$; ϕ_{M} = 'names of municipalities'; $\phi_{\mathsf{A\text{-}A}}$ = 'addresses' and $\phi_{\mathsf{P\text{-}A}} = \phi_{\mathsf{A\text{-}A}} = \phi_{\mathsf{L\text{-}A}}$ (A-A is Supplier); $\phi_{\mathsf{A\text{-}N}}$ = 'numbers for apartments in a building' and $\phi_{\mathsf{P\text{-}N}} = \phi_{\mathsf{A\text{-}N}} = \phi_{\mathsf{L\text{-}N}}$ (A-N and L-N are Suppliers); ϕ_{C} = 'fiscal categories of apartments'; ϕ_{A} = 'areas of apartments in square meters'; ϕ_{T} = 'amounts paid for local tax'. For role names it is: ρ_{O} = 'people' and $\rho_{\mathsf{O}} = \rho_{\mathsf{P}}$; ρ_{M} = 'apartment for the Ministry of Finance'

and $\rho_{\mathsf{M}} = \rho_{\mathsf{P\text{-}A}} = \rho_{\mathsf{P\text{-}N}}$; $\rho_{\mathsf{A\text{-}A}}$ = 'apartment for the Municipality' and $\rho_{\mathsf{A\text{-}A}} = \rho_{\mathsf{A\text{-}N}}$; ρ_{C} = 'category'; ρ_{A} = 'area'; $\rho_{\mathsf{L\text{-}A}}$ = 'apartment for the Municipality Tax Office' and $\rho_{\mathsf{L\text{-}A}} = \rho_{\mathsf{L\text{-}N}}$; ρ_{T} = 'amount'.

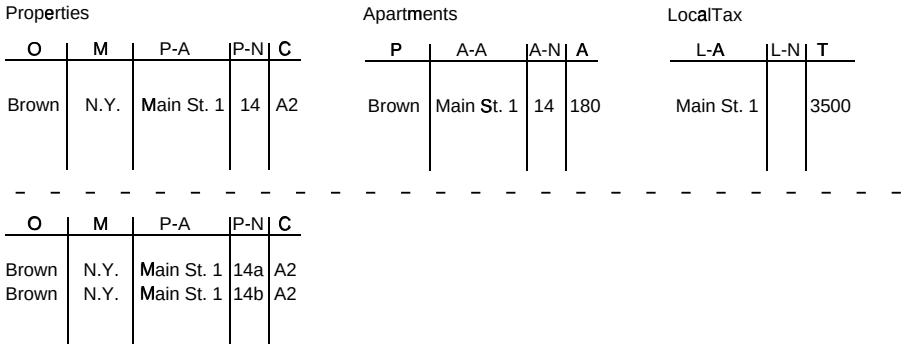

Fig. 1. Tables used in the running example: (a), top, before the update in the Ministry of Finance's DBMS, and (b), bottom, after the update

Interpretation. We call *interpretation* of a table a mapping from its extension to U associating each tuple t to an object of U which is (partially) described by values in t. The mapping between tuples of r_a and objects of U is materialized by a range function $\bar{\rho}_a$, introduced in the following definition, whose purpose is to provide for each tuple $t \in r_a$ the element $\bar{\rho}_a(t)$ of the universe of the discourse whose value of feature ϕ_a is represented by $t.a$.

Definition 1. *Given the signature* $\langle \Phi, \mathrm{P}, \mathcal{F}, \mathcal{R}, S \rangle$ *of* $\Sigma(R, A)$ *an* interpretation *is provided by the couple* $\langle U, \mathbf{E} \rangle$ *where* U *is any set, denoting the reality of interest and* $\mathbf{E}$ *is a family of* range functions $\bar{\rho}_a : \mathrm{ext}(r_a) \mapsto U$, *one for each role* ρ_a.

We note that in general, to a relation r, through its attributes, more than one role is associated. Correspondingly, we have in general more than one range function associated to the same relation. Note that since two or more attributes, in the same relation, may share the same role then they share the same range function.

For the sake of explanation we use $\phi(x)$ to indicate the value of feature ϕ for element $x \in U$. We say that a relation r *represents* an element $x \in U$ if $\exists t \in \mathrm{ext}(r)$ such that for an attribute $a \in r$ it is $\bar{\rho}_a(t) = x$. Note that if a correctly represents $\phi(x)$ then it will exist (at least) a tuple $t \in \mathrm{ext}(r_a)$ such that $\bar{\rho}_a(t) = x$ and $t.a = \phi(x)$.

Example 2. In the relation Properties(O, M, P-A, P-N, C) it exists a mapping from tuples to the (class of) objects of U which are persons and whose feature 'names of people' is represented by values of O and a distinct mapping to the (class of) objects of U which are apartments and whose feature 'addresses' is

represented by values of P-A. For range functions we only consider in our example $\bar{\rho}_{\mathsf{O}} : \text{ext}(\mathsf{Properties}) \mapsto U$ and $\bar{\rho}_{\mathsf{P}} : \text{ext}(\mathsf{Apartments}) \mapsto U$ that map tuples to persons in U, $\bar{\rho}_{\mathsf{M}} : \text{ext}(\mathsf{Properties}) \mapsto U$ and $\bar{\rho}_{\mathsf{A\text{-}A}} : \text{ext}(\mathsf{Apartments}) \mapsto U$ that map tuples to apartments in U. Note that $\bar{\rho}_{\mathsf{M}} = \bar{\rho}_{\mathsf{P\text{-}A}} = \bar{\rho}_{\mathsf{P\text{-}N}}$ and $\bar{\rho}_{\mathsf{A\text{-}A}} = \bar{\rho}_{\mathsf{A\text{-}N}}$.

An example of the mapping defined by the above range functions is depicted in figure 2.

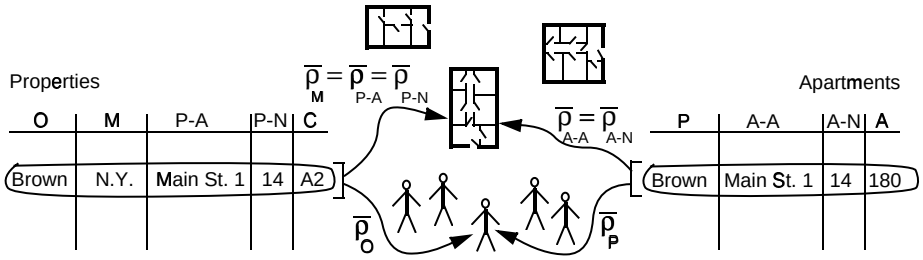

Fig. 2. An example of range functions

2.2 Coherence

We can now provide definitions needed to characterize the domain of a feature and to specify the meaning of coherence.

Definition 2. *Given a subset $B \subseteq A$, we let $fset_{\phi,B} : U \mapsto 2^D$ denote the* feature-set *of feature ϕ for element x with respect to B as:*

$$fset_{\phi,B}(x) = \{y \in D \mid \ a \in B_\phi,\ t \in \text{ext}(r_a),\ \bar{\rho}_a(t) = x,\ t.a \neq \mathsf{null},\ t.a = y\}.$$

Please note that while we admit null as one of the possible values in D we take it out from the definition of the domain of a feature.

Example 3. With reference to the situation shown in figure 1 if x denotes the apartment number 14 owned by Mr. Brown and (incoherently) represented in relation Properties after the update (bottom) as the apartment number $14a$, we have $fset_{\text{'numbers for apartments in a building'},\{\mathsf{P\text{-}N},\,\mathsf{A\text{-}N}\}}(x) = \{14, 14\mathsf{a}\}$.

We stress that an attribute $a \in A$ represents a value $\phi(x)$ for feature $\phi \in \Phi$ of an element $x \in U$ if it contributes a value to the feature-set of ϕ for x, i.e., if $|fset_{\phi,\{a\}}(x)| = 1$. Moreover, when $B_\phi \neq \emptyset$ (for instance this condition is satisfied for every $B \supseteq S$, as there has to be a supplier for each feature ϕ), it is $fset_{\phi,B}(x) = \emptyset$ if either (i) x is not represented by any relation containing attribute $a \in B_\phi$ or (ii) x is represented in some r_a such that $a \in B_\phi$, but $\phi(x)$ is not represented in any such a tuple, i.e., $t.a = \mathsf{null}$ for every tuple $t \in \text{ext}(r_a)$ such that $\bar{\rho}_a(t) = x$.

We give two definitions for coherence. The motivation for having both types of coherence is that in many cases it is useful to allow users to have each its own representation for $\phi(x)$. A weaker synchronization condition makes this possible in the case no supplier represents $\phi(x)$ either. The second definition is stricter: we have a strongly coherent representation for feature ϕ only if, for every $x \in U$, at least one supplier represents $\phi(x)$ each time at least one user does it. Moreover, all sources representing $\phi(x)$ have to agree on this value.

Definition 3. *We say* (R, A) *is* weakly coherent *if* $\Sigma(R, A)$ *is such that:* $\forall \phi \in \Phi,\ \forall x \in U,\ |fset_{\phi,S}(x)| > 0 \Rightarrow |fset_{\phi,A}(x)| = 1$.

Definition 4. *We say* (R, A) *is* strongly coherent *if it is weakly coherent and* $\Sigma(R, A)$ *is such that:* $\forall \phi \in \Phi,\ \forall x \in U,\ |fset_{\phi,S}(x)| = 0 \Rightarrow |fset_{\phi,A}(x)| = 0$.

Definition 3 formalizes a natural 'relaxed synchronization' condition: a feature ϕ is represented in a weakly coherent way if for every element $x \in U$ either no supplier represents $\phi(x)$ or all those sources representing it store the same value. If no supplier represents $\phi(x)$ then we impose no constraint on the representation of $\phi(x)$ among the users. That is, if $fset_{\phi,S}(x) = \emptyset$ then AKS will tolerate that users are not synchronized and will not provide a value for feature ϕ of element x. On the other side, by Definition 4, strong coherence still admits a supplier of ϕ refers to x but has not a representation of $\phi(x)$ as long as there is some supplier of ϕ providing a representation for $\phi(x)$. Moreover, all representations have to agree. Note that Definition 4 is given so that if $\Sigma(R, A)$ is strongly coherent then a 'contractual obligation' is enforced on the set of suppliers, in the sense it is forbidden that all of them store for element x a null for $\phi(x)$ if there is at least one user which represents $\phi(x)$.

Example 4. If we consider tables in figure 1 under a weak coherence assumption, since A-N in Apartments is one of the Supplier attributes for $\phi_{\text{A-N}} = \phi_{\text{P-N}} = \phi_{\text{L-N}}$, if the tuple t considered in Apartments had $t.\text{A-N} = \text{null}$ then in Properties (after the update, bottom in the figure) the two tuples with values 14a and 14b did not cause any incoherence. If we now consider the three tables before the update (top in the figure) in a strong coherence framework then it is not allowed that any of Apartments and LocalTax, both Suppliers, has a tuple representing feature values for the apartment considered in the running example and none of them provides a value ($\neq$ null) for the feature 'numbers for apartments in a building' for this apartment.

It is worthwhile to stress that the way coherence has been defined does not imply that two distinct source databases representing a same feature have to represent the same subset of U. More precisely, for distinct attributes $a_1, a_2 \in A_\phi$, an AKS does not require that an element $x \in U$ represented by r_{a_1} has to be also represented by r_{a_2}.

We finally remark that since in the above definitions sets of suppliers and users are parametric, this makes it possible to more easily deal with the initial transition phases towards coherence when many organizations are involved. In

such a context, in fact, one can start by choosing initially a very restricted set of suppliers and letting $A = S$. It is then possible to incrementally enlarge both A and S towards the actual situation by checking the state of coherence of the overall system at each step.

3 Implementation

The proposed model, as it has been discussed until now, is still completely based on semantic knowledge, represented by features, roles, and range functions. From an operational point of view, we need an implementation of these concepts that allows to efficiently check if two, or more, source databases together provide a coherent view of the universe of the discourse or not.

3.1 Dealing with the universe of the discourse

According to definitions introduced up to this point, the process of materializing an AKS would require a full knowledge of range functions $\bar{\rho}_a$ for each $a \in A$, that is a real knowledge of elements of the reality of interest. But the ultimate aim of an AKS is only the coherence maintenance for the considered set of features and set of suppliers. Hence we may leave the knowledge of features, roles, and range functions encapsulated within the Source Databases each a belongs to. In the end AKS only needs to know whether two given distinct tuples t_1 and t_2 speak about the same element of U or not. Of course, this reference to elements of U has to be considered in the context of roles played by attributes. For this process we then need a few additional concepts.

Definition 5. *Given two tuples $t_1, t_2 \in \mathrm{ext}(R)$ and two roles $\rho_1, \rho_2 \in \mathrm{P}$, we define the following predicate:*

$$\mathrm{idem}(t_1, t_2, \rho_1, \rho_2) = \begin{cases} \textit{true if} \quad \bar{\rho}_1(t_1) = \bar{\rho}_2(t_2) \quad \textit{in the case} \quad t_1 \in r_{\rho_1} \wedge t_2 \in r_{\rho_2} \\ \textit{false otherwise} \end{cases}$$

Let $I_\phi = \{\mathrm{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) \mid \mathrm{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) = \textit{true},\ a_1, a_2 \in A_\phi\}$ be the set of all true instances of *idem* predicate referring to the same feature ϕ.

For a feature ϕ, we can associate with I_ϕ a natural equivalence relation $\sim_\phi$ defined as:

$$\mathrm{idem}(t_1, t_2, \rho_1, \rho_2) \sim_\phi \mathrm{idem}(t'_1, t'_2, \rho'_1, \rho'_2) \Leftrightarrow \bar{\rho}_1(t_1) = \bar{\rho}'_1(t'_1)$$

partitioning I_ϕ into disjoint equivalence classes $[\mathrm{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2})] \in I_\phi / \sim_\phi$, one for each element $x \in U$ which is actually represented by some relation in $\mathrm{ext}(R_\phi)$, even if $\mathit{fset}_{\phi,A}(x) = \emptyset$.

Let $u_\phi : I_\phi / \sim_\phi \mapsto U$ be the (injective) function mapping equivalence classes into element of U as: $u_\phi([\mathrm{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2})]) = \bar{\rho}_{a_1}(t_1)$. Note that by definition an element $x \in U$ is represented in some relation $r \in R_\phi$ if and only if $x \in Im(u_\phi)$.

With the next definition we introduce a function providing, independently from the state of coherence of (R, A), the value of a given feature for a given element x of the universe of the discourse.

Definition 6. *We let $f_{\phi,B} : U \mapsto D$ denote the function providing the value of feature ϕ for element x as:*

$$f_{\phi,A}(x) = \begin{cases} y & \text{if } fset_{\phi,A}(x) = fset_{\phi,S}(x) = \{y\} \\ \mathsf{null} & \text{if } fset_{\phi,A}(x) = \emptyset \ \wedge \ x \in Im(u_\phi) \\ \bot & \text{if } x \notin Im(u_\phi) \text{ (and, obviously, } fset_{\phi,A}(x) = \emptyset) \\ \lambda & \text{if } fset_{\phi,S}(x) \neq \emptyset \ \wedge \ |fset_{\phi,A}(x)| \geq 2 \\ \lambda_w & \text{if } fset_{\phi,S}(x) = \emptyset \ \wedge \ |fset_{\phi,A \setminus S}(x)| \geq 1 \end{cases} .$$

We stress that, if $fset_{\phi,A}(x) = \emptyset$, the function u_ϕ mapping equivalence classes into elements of U allows to distinguish the case (i) $f_{\phi,A}(x) = \bot$, in which x is not represented by any relation in R_ϕ, from (ii) $f_{\phi,A}(x) = \mathsf{null}$, where $t.a_i = \mathsf{null}$ for each $a_i \in A_\phi$ such that there is a tuple $t \in \mathrm{ext}(r_{a_i})$ (at least one) with $\bar{\rho}_{a_i}(t) = x$. Moreover, we note that: (i) independently from the kind of coherence, $\phi(x)$ is incoherently represented if $f_{\phi,A}(x) = \lambda$, as there are at least two attributes representing $\phi(x)$ with distinct values and at least one is a supplier, while (ii) in the case of strong coherence, $\phi(x)$ is incoherently represented also in the case there is no supplier representing $\phi(x)$ and at least an user represents it.

As a direct consequence of Definitions 3, 4 and 6 we can now state following propositions, formulated in terms of the function $f_{\phi,A}$.

Proposition 1. (R, A) *is* weakly coherent *if and only if* $\Sigma(R, A)$ *is such that:* $\forall \phi \in \Phi, \ \forall x \in U \quad f_{\phi,A}(x) \neq \lambda$.

Proposition 2. (R, A) *is* strongly coherent *if and only if* $\Sigma(R, A)$ *is such that:* $\forall \phi \in \Phi, \ \forall x \in U \quad f_{\phi,A}(x) \notin \{\lambda_w, \lambda\}$.

Given the above results, it is clear that all we need to make it work is to represent all the equivalence classes induced by true *idem* predicates. This allows to retrieve tuples in various Source Databases referring to the same elements of the universe of the discourse and makes it feasible to check their coherence with respect to the representation of feature values.

Finally, note that the correctness of an attribute value is not something directly dealt with in the definition and implementation of an AKS. Suppliers for a given attribute are the only ones responsible for correctness, while an AKS is only used in dealing with the maintenance of coherence among relations supplying values and relations using them.

3.2 Materialization

The additional knowledge of understanding which are the true *idem* predicates, hence how are equivalence classes formed, knowledge that allows to materialize an AKS, is not necessarily present in any of the Source Database and has to be provided during the implementation phase. Note that such a knowledge is extensional and of semantic nature since it allows to say that two elements x_1 and x_2 in U, retrieved by means of two, generally independent, range functions $\bar{\rho}_1$ and $\bar{\rho}_2$, are the same element of U. Hence, even if automated tools can be

used to deal with the bulk of such correspondences [16], in our framework we assume that human beings have the task to understand when $\bar{\rho}_1(t') = \bar{\rho}_2(t'')$, and decide whether $\text{idem}(t', t'', \rho_1, \rho_2)$ is true or not, in cases unresolvable by automatic scrutiny.

Furthermore, note that since this knowledge is additional and needs not to be stored within any of the Source Databases then no change to any schema or extension in any of the Source Databases is required for such an implementation. This is really important from an organizational point of view, given the large autonomy the various organizations involved in the cooperation have.

Before presenting the mechanism we introduce to materialize such extensional knowledge we need to introduce two different paradigms to deal with coherence at the implementation level, namely, the *passive* paradigm and the *active* one:

- in the *passive paradigm* materialization is executed only for elements of U represented in at least two distinct relations such that at least one is a supplier;
- in the *active paradigm* materialization is also executed in the case an element of U is represented in just one relation.

Note that in the case of strong coherence the active paradigm is mandatory. Otherwise there would be no means to distinguish the case when just one supplier is representing $\phi(x)$, which is allowed, from the case when just one user is representing $\phi(x)$, which is not allowed and requires that also a supplier represents $\phi(x)$. Both cases, in fact, are equivalent in the passive paradigm and materialize no tuples for x.

We recall that the fact that attributes a_1 and a_2 in two distinct relations r_1 and r_2, respectively, represent the same feature does not imply the two relations share in their extension a representation of the same portion of the universe of the discourse. But the consequence of this fact is that for a tuple $t_1 \in r_1$ it may be $\text{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) = \mathit{false}$ for each $t_2 \in r_2$. In other words, the element of U tuple t_1 makes reference to has no representation in r_2.

To represent the extensional knowledge needed for coherence maintenance, we introduce the *Access Keys Data Base* (shortly, AKDB). This is a database whose scheme contains two relation schemes for each feature whose representation we want to keep synchronized in Source Databases. Using these relations, called *synchronization* and *identity*, we can now materialize and store values for the various equivalence classes of *idem* predicates and we can represent in AKDB the fact that an AKS is weakly or strongly coherent. The extension of AKDB thus contains the extensional knowledge needed to deal with coherence maintenance. Let sr_a be the sub-relation of r_a obtained by considering all the attributes of r_a having the same role of a (this included). Namely, sr_a is a relation whose scheme is $[b_1, b_2, \ldots]$ where each b_i is such that $b_i \in r_a$ and $\rho_{b_i} = \rho_a$ and whose extension is $\Pi_{b_1, b_2, \ldots}\, r_a$. Let $t[sr_a]$ denote the restriction of a tuple $t \in \text{ext}(r_a)$ to the sub-relation sr_a, i.e., $t[sr_a] \in \Pi_{b_1, b_2, \ldots}\, r_a$ and $t.b_i = t[sr_a].b_i$, for each $b_i \in sr_a$.

For shortness, we will discuss only the *maintenance of weak coherence in the case of passive paradigm*. Please note that the case of active paradigm does not require the introduction of additional concepts.

The synchronization relation. The synchronization relation is defined in AKDB for each feature ϕ with the purpose of explicitly representing the state of coherence/incoherence for various representations of the same element of the universe.

Let $\widehat{K}_\phi = \{\text{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) \in I_\phi \mid \ a \in S, t_1 \neq t_2, t_1.a_1 \neq \mathsf{null}, t_2.a_2 \neq \mathsf{null}\}$.

Definition 7. *A* synchronization relation σ_ϕ *for feature* ϕ *has the scheme* $h \cup \{k_1, k_2, \ldots, k_f\}$, *where* $f = |A_\phi|$, h *is a superkey of* σ_ϕ, *and* k_i *is a superkey of* sr_{a_i}, *for every* $a_i \in A_\phi$. *The* extension of σ_ϕ *for feature* ϕ *contains one tuple* t *for each equivalence class in* $\widehat{K}_\phi / \sim_\phi$, *where* $t.a_1 = t_1.a_1$ *and* $t.a_2 = t_2.a_2$, *for every* $\text{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) \in [\text{idem}(t, t', \rho, \rho')]$. *All remaining* $t.a_i$ *are set to* null.

Notice that it may be that $h \subseteq \{k_1, k_2, \ldots, k_f\}$ for a σ_ϕ. Indeed, the reason to explicitly denote with h a superkey for σ_ϕ is that it may be the case that no single k_i or combination of them can be a superkey for σ_ϕ. This happens, for example, whenever values for a given feature are partitioned according to the various a_i. In such a case for every $t_1 \in \text{ext}(r_{a_1})$ and $t_2 \in \text{ext}(r_{a_2})$ it is always $t_1.a_1 \neq t_2.a_2$, hence for every $t \in \text{ext}(\sigma_\phi)$ it is either $t.a_1 = \mathsf{null}$ or $t.a_2 = \mathsf{null}$ or both: clearly in this case it may be possible that no superkey for the synchronization relation can be built by using only attributes in $\{k_1, k_2, \ldots, k_f\}$. If this is the case then h, which is a superkey managed internally to AKDB, makes it possible the unique identification of tuples in σ_ϕ.

We note that if σ_ϕ synchronizes $f > 2$ attributes, the same tuple of σ_ϕ may represent up to $\binom{f}{2}$ true instances of *idem*. This is an upper bound and not the actual number since, given an element x of U for which a true instance of *idem* exists in I_ϕ, not necessarily all the roles involved in I_ϕ have to refer to x. In other words, it may happen that while r_1 and r_2 store $\phi(x)$ respectively in $t_1.a_1$ and $t_2.a_2$, a relation r_3 representing values for the same feature does not have in its representation of the reality of interest a tuple referring to x. In such a case σ_ϕ will contain a tuple t such that $t.a_1 = t.a_2 = \phi(x)$ and $t.a_3 = \mathsf{null}$.

The identity relation. It is important to note that it makes sense to check and represent coherence/incoherence only for tuples in the source database which, beyond referring to the same element of the universe, represent its values according to the considered paradigm. This is not always the case. For example, if a value $\phi(x)$ is represented only by two users, it does not satisfy the requirement of the passive paradigm, which requires for a value $\phi(x)$ to be represented by at least two attributes such that at least one is a supplier. In cases like this, it is anyhow useful to keep track of the fact that the considered tuples in source databases are referring to the same element, even if they cannot be represented in the synchronization relation, since they are "out-of-paradigm". We therefore introduce in AKDB a second relation γ_ϕ for each feature $\phi \in \Phi$ for this aim. This relation γ_ϕ, whose role is to contain tuples "out-of-paradigm", has the same scheme as σ_ϕ.

Definition 8. *The extension of the* identity relation γ_ϕ *for feature* ϕ *contains one tuple* t *for each equivalence class* $[\text{idem}(t, t', \rho, \rho')]$ *in* $(I_\phi / \sim_\phi) \setminus (\widehat{K}_\phi / \sim_\phi)$, *where* $t.a_1 = t_1.a_1$ *and* $t.a_2 = t_2.a_2$, *for each* $\text{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) \in [\text{idem}(t, t', \rho, \rho')]$. *All remaining* $t.a_i$ *are set to* null.

Notice that by definition it is $\text{ext}(\sigma_\phi) \cap \text{ext}(\gamma_\phi) = \emptyset$. For simplicity's sake, we denote as $\tau_{\phi,x}$ the only tuple belonging to $\text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$ which actually represents element $x \in Im(u_\phi)$.

It is important to stress that, as for Propositions 1 and 2, also Definitions 7 and 8 of synchronization and identity relations can be stated in terms of function $f_{\phi,A}$. In fact, given a feature $\phi \in \Phi$ and an element $x \in Im(u_\phi)$, one has:

- if $f_{\phi,A}(x) = \text{null}$, then $\tau_{\phi,x} \in \text{ext}(\gamma_\phi)$ and $\tau_{\phi,x}.a_i = \text{null}$, $\forall a_i \in \gamma_\phi$;
- if $f_{\phi,A}(x) = \lambda$, then $\tau_{\phi,x} \in \text{ext}(\sigma_\phi)$ and a state of incoherence is detected and recorded;
- if $f_{\phi,A}(x) = \perp$ then $\tau_{\phi,x} \notin \text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$, since x is not represented by any relation;
- if $f_{\phi,A}(x) = y$ then $\tau_{\phi,x} \in \text{ext}(\sigma_\phi)$ only if at least two attributes (one of them is necessarily a supplier) represent $\phi(x)$, $\tau_{\phi,x} \in \text{ext}(\gamma_\phi)$ otherwise;
- if $f_{\phi,A}(x) = \lambda_w$, then $\tau_{\phi,x} \in \text{ext}(\gamma_\phi)$;

4 Dynamics

We denote with $\Delta(\Sigma(R, A))$ (or simply Δ when no ambiguity arises) the materialization of the AKDB for a given $\Sigma(R, A)$. Δ allows the maintenance of coherence during updates to Source Databases trough a continuous exchange flow of information between AKDB and Source Databases. In fact, it receives the communication of the changes executed by each synchronized attribute (also said *change messages*) and sends a communication (referred as *incoherence message*) for each attribute whose value incoherently represents $\phi(x)$. So, by means of message-passing, AKDB allows to detect incoherence and to recover from it.

Without loss of generality, since features are synchronized in an independent way, we consider the case of a synchronization of a single feature ϕ. For the sake of simplicity we assume only one supplier for ϕ, denoted a_1.

Following notations are used in the rest of the paper:

- $\mu_\phi = (i, M_\phi, \tau_{\phi,x}.a_1, k_i)$ is the information attached to the incoherence message with index M_ϕ sent from AKDB to the Source Database containing r_{a_i};
- $\mu = (i, M_i, t[sr_{a_i}], t'[sr_{a_i}])$ is the information attached to the change message with regard to any kind of update in the value of the synchronized attribute a_i, where $t[sr_{a_i}]$ is the tuple before the change and $t'[sr_{a_i}]$ is the same tuple after the change. Parameter M_i indicates that this change is due to the incoherence message with index M_i if $M_i \neq 0$, while $M_i = 0$ denotes that a_i has independently changed the value of the synchronized attribute. We assume that $t'[sr_{a_i}] = \emptyset$ denotes a tuple deletion while $t[sr_{a_i}] = \emptyset$ denotes a tuple insertion.

We assume that Δ has been materialized in such a way that for each element x represented in some relation in R_ϕ, there is a tuple $\tau_{\phi,x}$ either in the identity relation γ_ϕ or in relation of synchronization σ_ϕ, according to whether $\tau_{\phi,x}$ is out-of-paradigm or not. Moreover, by following Proposition 1, an incoherence state is recorded for each (not out-of-paradigm) tuple $\tau_{\phi,x} \in \text{ext}(\sigma_\phi)$ such that $f_{\phi,A}(x) = \lambda$.

Before showing how the materialization of an AKDB for $\Sigma(R, A)$ allows the maintenance of coherence during changes to Source Databases, we first show how Δ is used when a change message $\mu = (i, M_i, t[sr_{a_i}], t'[sr_{a_i}])$ is received, that is if an attribute value is changed in some tuple, or what to do if a tuple is added or deleted:

- *Change of value.* We assume that both $t[sr_{a_i}].a_i$ and $t'[sr_{a_i}].a_i$ are not null, otherwise we are in cases similar to an insertion or a deletion of a tuple. Using μ we can access tuple $\tau \in \text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$ such that $\tau.k_i = t.k_i$ and change it so that $\tau.a_i = t'.a_i$.
- *Deletion of a tuple.* Using μ we can access tuple $\tau \in \text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$ such that $\tau.k_i = t.k_i$ and set $\tau.k_i = \text{null}$.
- *Insertion of a tuple.* The insertion of a tuple t' in r_{a_i} needs the identification of tuples representing the same element. So, a negotiation phase between sources, involving human beings, could happen to identify the tuple $\tau_{\phi,x} \in \text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$ associated to the element $x = \bar{\rho}_{a_i}(t')$. The negotiation has two possible outcome: (i) no materialized $\tau_{\phi,x}$ exists in $\text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$, or (ii) $\tau_{\phi,x}$ exists. In the first case, a tuple $\tau_{\phi,x}$ is inserted in $\text{ext}(\gamma_\phi)$. In both cases $\tau_{\phi,x}$ it is set $\tau_{\phi,x}.k_i = t'.k_i$.

Each kind of change previously analyzed is communicated from source databases by sending a change message μ to AKDB. From a practical point of view, a coherence test is actually performed after executing a set of changes on tuples in $\text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$, according to requirement in a sequence of change messages $\langle \mu_1, \mu_2, \ldots \mu_l \rangle$, where $\mu_j = \mu(i_j, x_{i_j}, t_{i_j}[sr_{a_{i_j}}], t'_{i_j}[sr_{a_{i_j}}])$ for $j = 1, \ldots, l$.

After having executed changes in the sequence, two tests must be executed to evaluate the coherence state of (R, A). As it is useless to check the state of coherence of tuples out-of-paradigm, before testing the coherence state of (R, A), it is necessary to check what tuples are out-of-paradigm after changes.

After these tests (executed in a blocking way), a set of incoherence messages μ_ϕ is sent for each $\tau_{\phi,x} \in \text{ext}(\sigma'_\phi)$ which is recorded as incoherent.

We stress that on accepting an incoherence message $\mu_\phi(1, k', \text{null}, k_1)$ a negotiation phase, involving human beings, between the supplier source and users representing $\phi(x)$ could happen, in order to determine the correct representation of $\phi(x)$.

Given the above described formal framework it can be proved that, under suitably defined temporal conditions, needed to ensure there is enough time between changes to allows for the system to enter in a steady state, the overall system moves from a state of overall coherence to a state of overall coherence.

Our approach therefore allows to follow an incremental route to coherence enforcement, and this is really needed, in real-life cases, to smoothly involve in the cooperation autonomous organizations, and hence to be successful in the coherence maintenance goals. Moreover, our approach allows to explicitly deal with technical and organizational costs of cooperation.

5 Conclusions

In this paper we have formalized and extended an architectural approach first sketched in [2] aiming at supporting data interoperability among legacy information systems of autonomous organizations. Our approach deals with the data coherency maintenance problem in the case of inter-organizational cooperation and provides methodological support for both phases of (i) detection of data inconsistency and (ii) recovery of incoherence once it is detected. More details on the dynamic aspects are presented in [8], while a formal computational model for the reference scenario of inter-organizational cooperation is sketched in [7].

Our approach allows to follow an incremental route to coherence enforcement, and this is really needed, in real-life cases, to smoothly involve in the cooperation autonomous organizations, and hence to be successful in the coherence maintenance goals. Moreover, our approach allows to explicitly deal, in large-scale and real-life contexts, with performance and organizational issues.

The formal framework presented has been validated through a number of inter-organizational cooperative information systems managed by the "Coordinamento dei Progetti Intersettoriali" of AIPA, the Italian Authority for Information Technology in Public Administrations and discussed in [3–6, 19].

References

1. S.Abiteboul: On views and XML, *SIGMOD Record*, 28(4):30–38, Dec.99.
2. F.Arcieri, E.Cappadozzi, P.Naggar, E.Nardelli, M.Talamo: Access Key Warehouse: a new approach to the development of cooperative information systems, *4th Int. Conf. on Cooperative Information Systems (CoopIS'99)*, Edinburgh, Scotland, U.K., 46–56, Sep.99. Extended version accepted for publication in the *Int. Journal of Cooperative Information Systems*.
3. F.Arcieri, C.Cammino, E.Nardelli, M.Talamo, A.Venza: The Italian Cadastral Information System: a Real-Life Spatio-Temporal DBMS, *Workshop on Spatio-Temporal Database Management (STDBM'99)*, Edinburgh, Scotland, U.K., Sep.99, Lecture Notes in Computer Science vol.1678, 79–99, Springer-Verlag.
4. F.Arcieri, E.Cappadozzi, E.Nardelli, M.Talamo: Distributed Territorial Data Management and Exchange for Public Organizations, *3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS'01)*, San Jose, Ca., USA, Jun.01, IEEE Computer Society Press, 2001.
5. F.Arcieri, E.Cappadozzi, E.Nardelli, M.Talamo: Geographical Information Systems Interoperability through Distributed Territorial Data Exchange, *International Workshop on Databases, Documents, and Information Fusion (DBFusion'01)*, Magdeburg, Germany, Apr.01.

6. F.Arcieri, E.Cappadozzi, E.Nardelli, M.Talamo: SIM: a Working Example of an E-Government Service Infrastructure for Mountain Communities, *Workshop on Electronic Government (DEXA-eGov'01)*, associated to the 2001 Conference on Databases and Expert System Applications (DEXA'01), Sept.01, Munich, Germany, IEEE Computer Society Press.
7. F.Arcieri, R.Giaccio, E.Nardelli, M.Talamo: A framework for inter-organizational public administration network services. *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR'01)*, L'Aquila, Italy, Aug.01. IEEE Computer Society Press, 2001.
8. F.Arcieri, G.Melideo, E.Nardelli, M.Talamo: On the Dynamics of an Infrastructural Approach Supporting Coherence Maintenance for Inter-Organizational Collaboration, *Software Trends, Open Space Symposium for Business, Education and Research - Business Strategy Based Software Engineering*, Sept.01, Vierwaldstttersee, Switzerland, NetAcademy Press.
9. D.Calvanese, G.De Giacomo, M.Lenzerini, D.Nardi, R.Rosati: A principled approach to data integration and reconciliation in data warehousing, *Int. Work. on Design and Management of Data Warehouses (DMDW'99)*, Heidelberg, Germany, 1999.
10. D.Calvanese, G.De Giacomo, M.Lenzerini, D.Nardi, R.Rosati, Information integration: conceptual modeling and reasoning support, *3rd Int. Conf. on Cooperative Information Systems (CoopIS'98)*, 280–291, 1998.
11. S.Conrad, B.Eaglestone, W.Hasselbring, M.Roantree, F.Saltor, M.Schönhoff, M.Strässler, M.W.W.Vermeer: Research Issues in Federated Database Systems: Report of EFDBS'97 Workshop, *SIGMOD Record*, 26(4):54–56, 1997.
12. S.Conrad, W.Hasselbring, U.Hohenstein, R.-D.Kutsche, M.Roantree, G.Saake, F.Saltor: Engineering Federated Information Systems: Report of EFIS'99 Workshop, *SIGMOD Record*, 28(3):9–11, 1999.
13. W.Hasselbring, W.-J. van den Heuvel, G.J.Houben, R.-D.Kutsche, B.Rieger, M.Roantree, K.Subieta: Research and Practice in Federated Information Systems: Report of the EFIS'2000 International Workshop, *SIGMOD Record*, 29(4):16–18, 2000.
14. W.Hasselbring: Information System Integration: introduction to the special section, *Communications of the ACM*, 43(6):33–38, June 2000.
15. V.Josifovski, T.Risch: Integrating Heterogeneous Overlapping Databases Through Object-Oriented Transformations, *25th Conf. on Very Large Data Bases (VLDB'99)*, 435–446, Edinburgh, Scotland, 1999.
16. M.A.Ouksel and A.P.Sheth: Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section, *SIGMOD Record*, 28(1):5–12, Mar.99.
17. E.A.Rundensteiner, A.Koeller, X.Zhang: Maintaining Data Warehouses over Changing Information Sources, *Communications of the ACM*, 43(6):57–62, June 2000.
18. M.Tork Roth, P.Schwarz: Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, *23rd Conf. on Very Large Data Bases (VLDB'97)*, 266–275, Athens, Greece, 1997.
19. M.Talamo, F.Arcieri, G.Conia, E.Nardelli: SICC: An Exchange System for Cadastral Information, *6th Int. Symp. on Large Spatial Databases (SSD'99)*, Hong Kong, China, Jul.99, Lecture Notes in Computer Science vol.1651, 360–364, Springer-Verlag.
20. Bulletin of the Technical Committee on Data Engineering, *Special Issue on Interoperability*, D.Kossmann (ed.), 21(3), September 1998.