On the Dynamics of an Infrastructural Approach Supporting Coherence Maintenance for Inter-Organizational Collaboration

F.Arcieri[1], G.Melideo[3], 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}@aipa.it

Abstract. In this paper we discuss further developments relative to an infrastructure supporting cooperation among independent organizations. In the realization of cooperative information systems one major obstacle is represented by the need of keeping coherence in the overall set of data needed for collaboration. On one side, in fact, data are independently and autonomously managed by the various organizations. On the other one, data are needed and used also outside the organization producing/managing them and controlling their changes.
The general framework introduced in papers [1, 5] for this purpose is here completed and formalized for what regards the dynamic aspects, that is the means by which incoherence can be recovered once it is detected. Our solution is rooted in experiences of development of inter-organization cooperative information systems managed by the "Coordinamento dei Progetti Intersettoriali" of AIPA, the Italian Authority for Information Technology in Public Administration.

Keywords: inter-organization cooperation, data interoperability, information systems integration.

1 Introduction

The application scenario we consider is made up by a set of autonomous and independent organizations that have to cooperate to reach common overall goals. Over the last couple of years, given the explosion of Internet connections, the importance of such a scenario has reached a peak and the topic of integration of Information Systems has become a really hot one [10].

An important characteristic of this scenario is that the involved organizations share a common semantic background and are available to make some of

their resources available to support cooperation. This means that it is possible to deal with and to solve many of the hardest interoperability problems existing at semantic level (e.g., understanding if two relational tables describe the same portion of the universe of the discourse or not) through human conducted negotiations and discussions in the definition and design phases of the target system. But the sharing of a common semantic background does not imply at all that the various representations within the cooperating organizations of the (shared) reality of interest have the same expression. In other words, different organizations have different representation models for the same 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.

Our focus is instead the *coherence of the overall (distributed) set of data.* On one side, in fact, data are independently and autonomously managed by the various organizations. On the other one, data are needed and used also outside the organization producing/managing them and controlling their changes. These clashing situations will produce incoherence in the overall set of data, sooner or later, with absolute certainty.

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.

The availability of the involved organizations does not mean, on the other side, that their resources for cooperation are unlimited. On the contrary, it is important for each organization involved to be able to understand and to evaluate the steady-state costs to support such an inter-organization cooperation. In fact, to keep one's own services up-to-date in a cooperative framework has technical and organizational costs. Our approach makes such costs visible and supports their negotiation in a flexible way. Hence its major organizational impact is to bring to the surface the hidden costs of inter-organization cooperation.

A recent workshop on engineering federated database systems [9] 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.
One widely followed approach to meet the above discussed goal is based on the design of a suitable Data Warehouse [11, 20, 22] and it is widely known that data integration is at the heart of data warehousing [12, 13].

We already introduced in [1, 5] a novel architectural approach to deal with these issues, based on the use of a suitable Data Warehouse in a new conceptual role and called the **Access Keys Warehouse** approach. Systems SICC [2, 15, 18, 19] for cadastral data exchange among italian Municipalities, Ministry of Finance, Notaries and Certified Land Surveyors, SCT [3, 4] for territorial data exchange among Public Administration, and SIM [6] for providing e-government

services to people living in mountain areas, have all been designed and implemented according to this approach.

In this paper we proceed further and formalize the dynamic aspects of the proposed infrastructure, showing how it allows to control the evolution of the overall system.

2 An example of incoherence generation

The following example, taken from [2] and referring to the management of cadastral data in Italy, shows a typical interaction among cooperating organizations leading to incoherence in the overall set of data. We shall use the relational tables involved as a running example to clarify how formal concepts are applied.

A Certified Land Surveyor prepares for a client a request for a variation to an apartment (e.g., to divide a large apartment in two smaller ones). The request is composed by some descriptive data and some geometric data and is stored in a database in the surveyor's office.
The Surveyor prints the request and send it by registered mail to the pertinent cadastral office of the Ministry of Finance. The office, having checked that everything has been done according to current laws and that data are coherent with data stored in cadastral databases executes the update.
The Municipality in whose competence territory the apartment is located in has an interest in knowing such a change, for local tax reasons (e.g., the two smaller apartments are different subjects, from a fiscal point of view, than the previous one). The Certified Land Surveyor has on obligation to get an approval for the change from the Building Service of the Municipality before submitting the request to the Cadaster.
Of course, until the latter one receives the request the change has not really happened. But neither the cadastral office nor the Surveyor have any legal obligation to inform the Municipality when the change really happens, i.e. when the request has been accepted by the Cadaster. This is the duty of the owner of the apartment and if he/she forgets to comply with this obligation, the Municipality may never be aware of the change until an inspector is sent in the apartment to check the situation and the lack of coherence is detected.

In terms of the underlying databases, the involved tables are Properties(O, M, P-A, P-N, C) in a Ministry of Finance's DBMS and Apartments(P, A-A, A-N, A) in a Municipality's DBMS. See in figure 1-top 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. After the above described sequence of events, situation will be the one depicted in figure 1-bottom 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. But the situation shown in figure 1-bottom has the potential of producing an additional, of different kind, and harder to check incoherence. In fact, it is the Municipality that defines the rules for assigning numbers to

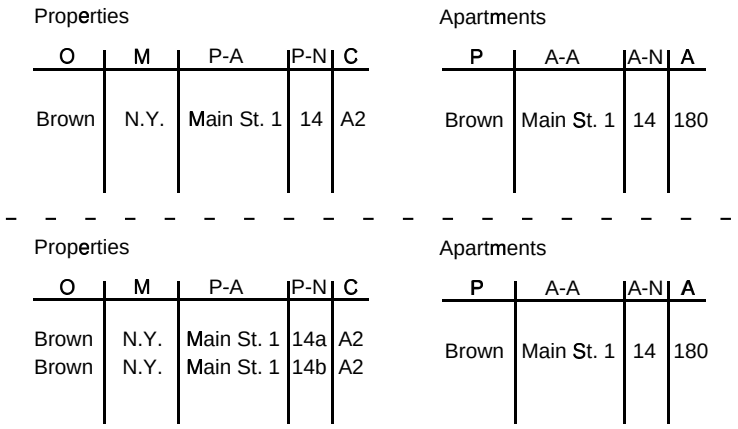

Fig. 1. Tables used in the running example

apartments in buildings and such rules require the new number is one plus the old highest number in the building.

Hence, when the Municipality receives the communication of the change, if this does not contain the numbers of the new apartments, then it may update its DBMS using the correct numbering schema and giving thus rise to the incoherence shown in figure 2. Consider that such an incoherence may remain unnoticed for a long time, e.g. until an inspector is sent on the place to check the situation.

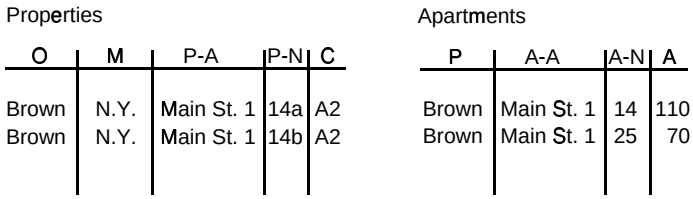

Fig. 2. An additional kind of incoherence

If, on the contrary, the communication of the change contains the numbers of the new apartments then the Municipality will try to have the Surveyor and the Cadaster and the Federal Agency to change their databases according to such a regulation. But since most probably these cadastral databases will already have been updated by then and since this issue of apartment numbering is not something the Cadaster has, by the law, to really care about, no action will be taken and the incoherence will remain there.

Note, from an organizational viewpoint, that such a mistake may have been unnoticed or unchecked in the prior request for approval submitted from the surveyor to the Municipality. In fact, the Building Service of the Municipality is not the one in charge of such a check on apartment numbering (the Toponymy

Service is in charge) and regulations require that the submission of the change request to the Cadaster only needs the approval of the Building Service.

2.1 Distributed coherence and related work

A widely followed approach to issues regarding data integration in multidatabase system is based on the *wrapper-mediator* architecture [22] coupled with an object-oriented approach (e.g., [8]). But the main drawback of such an approach in large-scale and legacy systems like the ones found in the scenario above described is that to consider the existing systems as black boxes may be catastrophic in terms of performances. In such a case, in fact, given the wrapping provided by the OO technology, access number and access paths required to an underlying Source Database by the execution of coherence maintenance functions are largely out of the control of the designer. Thus, providing acceptable performances is a highly challenging task [14].

Our approach, on the contrary, makes it possible to evaluate and tune the impact on performances of a given Source Database deriving from outside requests. Hence our approach makes it possible to perform a rightsizing of the overall system through a cost-benefit analysis.

3 The formal model

We assume that the cooperating organizations have their data stored in relational databases. Difficulties of dealing with non-relational technology are tackled by using *wrappers* [17, 21] 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. Each feature value is taken from a universe D of values. Relations in the various Source Databases represent (elements of) U by storing values for the features of the elements of U that are more relevant to the modeled fragment of the reality of interest as values of their attributes.

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. 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.

The Access Keys Warehouse 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.

We briefly recall the main elements of the formal model, described in fuller detail in [5], and explain them by means of a running example.

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.
For an attribute $a \in A$ we write $r_a \in R$ to denote the relation containing a (written also as $a \in r_a$). For a relation $r \in R$ we write $\text{ext}(r)$ to denote the set of tuples belonging to the extension of r and we let $\text{ext}(X) = \cup_{r \in X} \text{ext}(r)$, for each $X \subseteq R$. Given $a \in r$ and $t \in \text{ext}(r)$, we denote with $t.a$ the value taken by tuple t in correspondence with a. Clearly, $t.a$ 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*.

3.1 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, denoted as ϕ_a;
- $\mathcal{R} : A \mapsto \mathrm{P}$ is the function providing for each attribute a in A the unique role name $\mathcal{R}(a)$ associated to it, denoted as ρ_a;
- $S \subseteq A$, denotes the set of *suppliers* for attribute values, $A \setminus S$ is called the set of *users*.

Soundness conditions for $\Sigma(R, A)$ require that attributes sharing the same role have to belong to the same relation and that there has to be a supplier for each feature.

Example 1. In our running example we have for features: $\phi_{\mathsf{O}} = \phi_{\mathsf{P}} =$ 'names of people', $\phi_{\mathsf{M}} =$ 'names of Local Agencies', $\phi_{\mathsf{P\text{-}A}} = \phi_{\mathsf{A\text{-}A}} =$ 'addresses' (and A-A is Supplier), $\phi_{\mathsf{P\text{-}N}} = \phi_{\mathsf{A\text{-}N}} =$ 'numbers for apartments in a building' (and A-N is Supplier), $\phi_{\mathsf{C}} =$ 'fiscal categories of apartments', and $\phi_{\mathsf{A}} =$ 'areas of apartments in square meters'. For roles it is: $\rho_{\mathsf{O}} = \rho_{\mathsf{P}} =$ 'people', $\rho_{\mathsf{M}} = \rho_{\mathsf{P\text{-}A}} = \rho_{\mathsf{P\text{-}N}} =$ 'apartment for the Federal Agency', $\rho_{\mathsf{A\text{-}A}} = \rho_{\mathsf{A\text{-}N}} =$ 'apartment for the Local Agency', $\rho_{\mathsf{C}} =$ 'category', and $\rho_{\mathsf{A}} =$ 'area'.

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 $R_\phi = \{r_a \mid a \in A_\phi\}$.

3.2 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. Each table r has, in general, more than one interpretation, since more than one mapping between tuples in r and objects in U exists.

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.

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,*
- $\mathbf{E}$ *is a family of* range functions $\bar{\rho}_a : \mathrm{ext}(r_a) \mapsto U$*, one for each role* ρ_a*.*

The purpose of the *range functions* in $\mathbf{E}$ 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 $t.a$. 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.

Example 2. In 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 $\bar{\rho}_{\mathsf{O}}$: ext(Properties) $\mapsto U$ and $\bar{\rho}_{\mathsf{P}}$: ext(Apartments) $\mapsto U$ that map tuples to person in U, $\bar{\rho}_{\mathsf{M}} = \bar{\rho}_{\mathsf{P\text{-}A}} = \bar{\rho}_{\mathsf{P\text{-}N}}$: ext(Properties) $\mapsto U$ and $\bar{\rho}_{\mathsf{A\text{-}A}} = \bar{\rho}_{\mathsf{A\text{-}N}}$: ext(Apartments) $\mapsto U$ that map tuples to apartments in U. You can see in figure 3 an example of the mapping defined by the above range functions. Note also that since two or more attributes, in the same relation, may share the same role then they share the same range function.

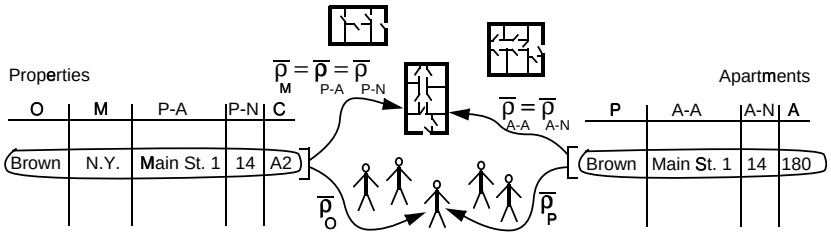

Fig. 3. An example of range functions

4 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:*

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

Let us 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.

Since in many cases it is useful to allow to users to have each its own representation for feature ϕ of element x (denoted $\phi(x)$), we give two definitions for coherence. 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:*

$$\forall \phi \in \Phi \quad \forall x \in U \quad |fset_{\phi,S}(x)| > 0 \Rightarrow |fset_{\phi,A}(x)| = 1.$$

Definition 4. *We say (R, A) is* strongly coherent *or simply* coherent *if it is weakly coherent and $\Sigma(R, A)$ is such that:*

$$\forall \phi \in \Phi \quad \forall x \in U \quad |fset_{\phi,S}(x)| = 0 \Rightarrow |fset_{\phi,A}(x)| = 0.$$

Please note that if (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 3. If we consider the two tables in figure 1-bottom under a weak coherence assumption, since A-N in Apartments is the Supplier attribute for $\phi_{\mathsf{A\text{-}N}} = \phi_{\mathsf{P\text{-}N}}$, if the tuple t considered in Apartments had $t.\mathsf{A\text{-}N} = \mathsf{null}$ then in Properties the two tuples with values 14a and 14b did not cause any incoherence.

Let us now add to our running example one more table in a Municipality's DBMS, namely LocalTax(L-A, L-N, T), with $\phi_{\mathsf{L\text{-}A}}$ = 'addresses' ($= \phi_{\mathsf{P\text{-}A}} = \phi_{\mathsf{A\text{-}A}}$), $\phi_{\mathsf{L\text{-}N}}$ = 'numbers for apartments in a building' ($= \phi_{\mathsf{P\text{-}N}} = \phi_{\mathsf{A\text{-}N}}$ and L-N is Supplier), and ϕ_{T} = 'amounts paid for local tax'. For roles it is $\rho_{\mathsf{L\text{-}A}} = \rho_{\mathsf{L\text{-}N}}$ = 'apartment for the Local Tax Office' and ρ_{T} = 'amount'. Range functions are accordingly defined. If we now consider the three tables 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 (see also figure 4).

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.

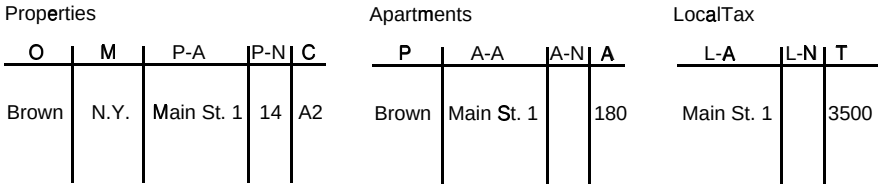

Fig. 4. A weakly coherent set of tables showing strong incoherence

5 Dealing with the universe of the discourse

This section presents implementation mechanisms of concepts introduced until now and still completely based on semantic knowledge (represented by features, roles, and range functions) that allow to efficiently check if two, or more, databases together provide a coherent view of the universe of the discourse.

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:*

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

Let us consider the set I_ϕ of all true instances of *idem* predicate referring to the same feature ϕ. Given an arbitrary feature ϕ, we can associate with the set I_ϕ a natural equivalence relation $\sim_\phi$. Note that, by definition, for every feature $\phi \in \Phi$, the relation $\sim_\phi$ partitions the set 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 $\mathrm{ext}(R_\phi)$, also if it is $fset_{\phi,A}(x) = \emptyset$.

Let $u_\phi : I_\phi \mapsto U$ 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 $x \in Im(u_\phi)$.

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

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

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

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:*

$$\forall \phi \in \Phi,\ \forall x \in U \quad f_{\phi,A}(x) \notin \{\lambda_w, \lambda\}.$$

All we need to make this approach works is to represent all the *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.

6 Materialization

The additional knowledge of understanding which are the true *idem* predicates, 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.

Before presenting the mechanism we introduce to materialize such extensional knowledge it is important 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 the universe of discourse 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 Source Database.

In order to explain the method which allows to implement the extensional knowledge needed for coherence maintenance, we introduce the following notations:

- $\widehat{I}_\phi \subseteq I_\phi$ is the set of all true instances of *idem* in which at least one of values $t_1.a_1$ and $t_2.a_2$ is not null,
- $\widehat{K}_\phi \subseteq \widehat{I}_\phi$ is the set of all true instances of *idem* referring to the same feature ϕ such that: the two tuples are distinct, at least one of the tuples refers to a supplier for ϕ, and both the values $t_1.a_1$ and $t_2.a_2$ are not null.

To represent such extensional knowledge we introduce the *Access Keys Data Base* (shortly, AKDB). This is a database whose scheme contains two relation schemes, namely the *synchronization relation* and the *identity relation* for each

feature whose representation we want to keep synchronized in Source Databases. Using these relations we can now materialize and store values for the various *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 $b_i \in sr_a$.

One synchronization relation is introduced in AKDB for each feature ϕ according to the following definition.

Definition 7. *A* synchronization relation σ_ϕ *for feature* ϕ *has the scheme* $h_\phi \cup \{k_1, k_2, \ldots, k_f\}$, *where* $f = |A_\phi|$, h *is a superkey of* σ_ϕ, *and* k_i *is a superkey of* sr_{a_i}, *for every* $a_i \in A_\phi$.

Definition 8. *The extension of the* synchronization relation σ_ϕ *for feature* ϕ *contains one tuple* t *for each element* $[\text{idem}(t, t', \rho, \rho')]$ *of*

- $\widehat{K}_\phi / \sim_\phi$, *in the case of* passive paradigm
- $\widehat{I}_\phi / \sim_\phi$, *in the case of* active paradigm

where $t.a_1 = t_1.a_1$ *and* $t.a_2 = t_2.a_2$, $\forall\ \text{idem}(t_1, t_2, \rho_{a_1}, \rho_{a_2}) \in [\text{idem}(t, t', \rho, \rho')]$. *All remaining* $t.a_i$ *are set to* null.

Example 4. In our running example we have, for the synchronization relation taking care of coherence of feature 'numbers for apartments in a building' among the three involved tables, the relation scheme CoNumbers(M, P-A, P-N, C, P, A-A, A-N, L-A, L-N). Notice that attributes C and P are present in this scheme only to provide better performances in the implementation. In fact, table Properties is horizontally partitioned, for efficiency reasons, according to the values of C (representing 'fiscal categories for apartments' feature's values), while tuples in table Apartments have an index defined on values of of P (representing 'names of people' feature's values).

The materialization of CoNumbers relation scheme (see figure 5) makes reference to the case presented in figure 2 and with the third relation subsequently added. The shown materialization assumes that human intervention has allowed to establish that tuples in the Local Agency's DBMS with $t.\text{A-N} = t.\text{L-N} = 14$ make reference to the element of U represented by the tuple in the Federal Agency's DBMS with $t.\text{P-N} = 14\text{a}$ (and similarly for the remaining reference).

We consider in AKDB a second relation γ_ϕ for each feature $\phi \in \Phi$, called the identity relation, with the same scheme as σ_ϕ, whose role is to contain tuples "out-of-paradigm".

Definition 9. *The extension of the* identity relation γ_ϕ *for feature* ϕ *contains one tuple for each element* $[\text{idem}(t, t', \rho, \rho')]$ *of*

CoNumbers

M	P-A	P-N	C	P	A-A	A-N	L-A	L-N
N.Y.	Main St. 1	14a	A2	Brown	Main St. 1	14	Main St. 1	14
N.Y.	Main St. 1	14b	A2	Brown	Main St. 1	25	Main St. 1	25

Fig. 5. Materialization of the synchronization relation for the running example

- $(I_\phi/\sim_\phi) \setminus (\widehat{K}_\phi/\sim_\phi)$, *in the case of* passive paradigm
- $(I_\phi/\sim_\phi) \setminus (\widehat{I}_\phi/\sim_\phi)$, *in the case of* active paradigm

where $t.a_1 = t_1.a_1$ *and* $t.a_2 = t_2.a_2$, $\forall$ idem$(t_1, t_2, \rho_{a_1}, \rho_{a_2}) \in$ [idem(t, t', ρ, ρ')]. *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 $x \in Im(u_\phi)$.

7 Dynamics

We denote with $\Delta(\Sigma(R, A))$ (or simply Δ when no ambiguity arises) the materialization of the AKDB for a given $\Sigma(R, A)$. $\Delta(\Sigma(R, A))$ allows the maintenance of coherence during updates to Source Databases trough a continuous exchange flow of information between Δ 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, Δ 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.

The following notations are used in the rest of the paper:

- M_ϕ is an integer value counting the number of incoherence messages sent from Δ to Source Databases (incremented each time an incoherence message is produced).
- for each synchronized attribute a_i:
  - X_i denotes an integer value counting the number of changes executed by a_i (incremented each time a change is occurred);
  - M_i denotes the index of the incoherence message about a_i most recently received by the Source Database containing r_{a_i}.
- $\mu_\phi = (i, M_\phi, \tau_{\phi,x}.a_1, k_i)$ denotes the information attached to the incoherence message with index M_ϕ sent from Δ to the Source Database containing r_{a_i}.

- $\mu = (i, X_i, M_i, t[sr_{a_i}], t'[sr_{a_i}])$ denotes the information attached to the change message with regard to any kind of update in the value of the synchronized attribute a_i, where X_i indicates that a_i executed its X_i-th change, $t[sr_{a_i}]$ is the tuple before the change, and $t'[sr_{a_i}]$ is the same tuple after the change. We assume that $t'[sr_{a_i}] = \emptyset$ denotes a tuple deletion while $t[sr_{a_i}] = \emptyset$ denotes a tuple insertion. 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 discuss both the cases of passive and active paradigms.

Definition 10. *Given a feature ϕ and an element $x \in Im(u_\phi)$, out-paradigm(ϕ, x) represents the following predicate:*

$$\textit{out-paradigm}(\phi, x) \equiv \begin{cases} f_{\phi,A}(x) \in \{\mathsf{null}, \lambda_w\} \vee \ (fset_{\phi,S}(x) \neq \emptyset \wedge fset_{\phi,A\setminus S}(x) = \emptyset) \\ \textit{in the case of passive paradigm} \\ \\ fset_{\phi,A}(x) = \emptyset \textit{ in the case of active paradigm} \end{cases}$$

For a feature ϕ and an element x which is represented by some relation in R_ϕ, we will say, for shortness, that tuple $\tau_{\phi,x}$ is *out-of-paradigm* if it is *out-paradigm(ϕ, x)*.

Definition 11. *$\forall x \in Im(u_\phi)$ such that $\neg$out-paradigm(ϕ, x), we define the predicate incoherent(ϕ, x) as:*

$$\textit{incoherent}(\phi, x) \equiv \begin{cases} f_{\phi,A} \in \{\lambda_w, \lambda\} \textit{ in the case of strong coherence} \\ f_{\phi,A} = \lambda \textit{ in the case of weak coherence} \end{cases}$$

We assume that $\Delta(\Sigma(R, A))$ 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, possible incoherences are detected and recorded. In particular, an incoherence state is recorded for each tuple $\tau_{\phi,x} \in \text{ext}(\sigma_\phi)$ such that *incoherent(ϕ, x)* is true.

7.1 Maintaining coherence in the overall system

The overall system may change in three possible ways: (i) a value may be changed in a tuple, (ii) an existing tuple may be deleted, (iii) a new tuple may be inserted.

Each kind of change is communicated from source databases through a sending of a change message μ to $\Delta(\Sigma(R, A))$. Let $\langle \mu_1, \mu_2, \ldots, \mu_l \rangle$ be a sequence of any type of updates previously analyzed involving attributes $a_i \in A_\phi$, where $\mu_j = \mu(i_j, x_{i_j}, t_{i_j}[sr_{a_{i_j}}], t'_{i_j}[sr_{a_{i_j}}])$ for $j = 1, \ldots, l$.

Although in asynchronous distributed systems message transmission delays are finite but unpredictable, we can assume without loss of generality that if $i_j = i_h$ and $j < h$ then $x_{i_j} < x_{i_h}$. That is, updates performed by the same

attribute $a_i \in A_\phi$ are assumed to be reported to AKDB by means of change messages μ in the same order in which they have been performed in r_{a_i}. In fact, from an operational point of view, a f-dimensional integer vector $V_\phi \in I\!N^f$ can record for each $i = 1, \ldots, f$ the index x_{i_j} associated to the last change performed on tuples in $\text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi)$ accordingly to the requirements in the sequence. So, on receiving a change message $\mu(i_j, x_{i_j}, t_{i_j}[sr_{a_{i_j}}], t'_{i_j}[sr_{a_{i_j}}])$, tuple $\tau_{\phi,x}.a_{i_j}$ is accordingly updated, being $x = \bar{\rho}_{a_{i_j}}(t) = \bar{\rho}_{a_{i_j}}(t')$, only if $V_\phi[i] < x_{i_j}$. On the contrary, if $V_\phi[i] > x_{i_j}$ the change would be ignored.

After having executed changes in the sequence, the following two tests must 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. Since correctness of tests can be guaranteed only if no update is executed during this phase, it is necessary for these tests to be performed in a "blocking" way, that is Δ is forced to wait until these two tests are executed, before receiving a new sequence of change messages.

Paradigm Test. Let $\text{ext}(\sigma'_\phi)$, $\text{ext}(\gamma'_\phi)$ denote respectively, the new set of tuple in the relations obtained after execution of this test. It is:

- $\text{ext}(\sigma'_\phi) = \{\tau_{\phi,x} \in \text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi) \mid \neg \mathit{out\text{-}paradigm}(\phi, x)\}$;
- $\text{ext}(\gamma'_\phi) = \{\tau_{\phi,x} \in \text{ext}(\sigma_\phi) \cup \text{ext}(\gamma_\phi) \mid \mathit{out\text{-}paradigm}(\phi, x)\}$.

Coherence Test. An incoherence state is recorded for each $\tau_{\phi,x} \in \text{ext}(\sigma'_\phi)$ such that $\mathit{incoherent}(\phi, x)$ holds.

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

- if $\tau_{\phi,x}.a_1 = \mathsf{null}$ then only the incoherence message $\mu_\phi = (1, M_\phi, \mathsf{null}, k_1)$ is generated;
- if $\tau_{\phi,x}.a_1 \neq \mathsf{null}$, then an incoherence message $\mu_\phi = (i, M_\phi, \tau_{\phi,x}.a_1, k_i)$ is generated for each a_i such that $\tau_{\phi,x}.a_i \neq \tau_{\phi,x}.a_1$.

We stress that on accepting an incoherence message $\mu_\phi(1, y, \mathsf{null}, k_1)$ 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 $\phi(x)$.

Let $\langle \mu_1, \mu_2, \ldots, \mu_l) \rangle$ be the sequence of incoherence messages received by source databases containing, respectively, r_{i_1}. Without loss of generality we can assume that incoherence messages related to the same synchronized attribute a_i are received in the same order in which they have been sent by $\Delta(\Sigma(R, A))$. This is possible as a variable M_i records the most recently received index of message. So, on receiving $\mu_\phi(i_j, y_{i_j}, a_{i_j}, k_{i_j})$, the message is ignored if $M_i > y_{i_j}$, while it is accepted if $M_i < y_{i_j}$. In this case, variable M_i is set to y_{i_j}.

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.

8 Open Problems

In order to complete the results contained in this paper, an interesting issue is to consider dynamic aspects dealing with the case of *multi-suppliers*, where more than one attribute may generate the value of a feature or may be entitled to change it.

As regards future work, with the increasing popularity of the Web, an interesting question could be to maintain coherence of information contained in a vast collection of semantically related web pages. As regards this issue, a formal model supporting this goal could be derived according to the solutions described in this and related papers [1–6] in the area of supporting coherence maintenance in the underlying legacy databases of cooperating organizations.

In the same context, when considering the area of e-government, the certification of exchanged e-services becomes of the utmost importance since very often exchanged data have a legal value and play a legal role. The main technical difficulty is that all e-services involved are, from the viewpoint of the certification process, like black boxes and cannot be internally changed. The only approach is therefore to monitor and to keep track of input and output flows which are nothing more than a sequence of IP packets. A preliminary investigation on the underlying computational model is reported in [7] and this issue will be the focus of our future research activity.

Acknowledgments. The authors want to thank E. Cappadozzi and P. Naggar for their contribution to the definition of the formal model.

References

1. F.Arcieri, E.Cappadozzi, P.Naggar, E.Nardelli, M.Talamo: Access Key Warehouse: a new approach to the development of cooperative information systems, *CoopIS'99*, U.K., Sep.99, accepted in the *Int. J. of Cooperative Information Systems*, 2001.
2. 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)*, U.K., Sep.99, LNCS 1678.
3. 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.

4. F.Arcieri, E.Cappadozzi, E.Nardelli, M.Talamo: Geographical information systems interoperability through distributed data exchange, *1st Int. Workshop on Databases, Documents, and Information Fusion (DBFusion'01)*, Germany, 2001.
5. F.Arcieri, E.Cappadozzi, G.Melideo, P.Naggar, E.Nardelli, M.Talamo: A formal model for data coherence maintenance, *International Workshop on Foundations of Models for Information Integration (FMII-2001)*, Viterbo, Italy, Sep.01, LNCS.
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)*, 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. O.Burkhes, A.Elmagarmid (eds.): *Object Oriented Multidatabase Systems*, Prentice Hall, 1996.
9. A.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*, 1997.
10. W.Hasselbring: Information System Integration: introduction to the special section, *Comm. of the ACM*, 43(6):33–38, June 2000.
11. R.Hull: Managing semantic heterogeneity in databases: a theoretical perspective, *(PODS'97)*.
12. W.H.Inmon: *Building the Data Warehouse*, John Wiley, 1996.
13. M.Jarke, M.Lenzerini, Y.Vassiliou, P.Vassiliadis: *Fundamentals of Data Warehouses*, Springer, 1999.
14. V.Josifovski, T.Risch: Integrating Heterogeneous Overlapping Databases Through Object-Oriented Transformations, *(VLDB'99)*, Edinburgh, 1999.
15. Il Catasto Telematico, *Notiziario Fiscale*, n.11-12, pp.19–22, Nov-Dec.98, Ministry of Finance, Roma.
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. M.Tork Roth, P.Schwarz: Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, *(VLDB'97)*, Athens, 1997.
18. M.Talamo, F.Arcieri, G.Conia, Il Sistema di Interscambio Catasto-Comuni (parte I), *GEO Media*, vol.2, Jul-Aug 1998, (parte II), *GEO Media*, vol.2, Sep-Oct 1998, Maggioli Editore, Roma (in italian).
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, Jul.99, LNCS 1651.
20. E.A.Rundensteiner, A.Koeller, X.Zhang: Maintaining Data Warehouses over Changing Information Sources, *CACM*, June 2000.
21. J.D.Ullman: Information integration using logical views, *(ICDT'97)*, LNCS 1186.
22. G.Wiederhold: Mediators in the architecture of future information systems, *IEEE Computer*, 1992.