Evolving Hybrid Distributed Databases: Architecture and Methodology

Philippe Thiran and Jean-Luc Hainaut

InterDB Project[1], Database Applications Engineering Laboratory
Institut d'Informatique, University of Namur, Belgium
{pth|jlh}@info.fundp.ac.be

Abstract. This paper focuses on the interoperability of autonomous legacy databases with the idea of meeting the future requirements of an organization. It describes a general architecture and a methodology intended to address the problem of providing new client applications with an abstract interface defined from the new requirements and the description of the legacy databases. The architecture relies on a hierarchy of mediators that provide the abstract interface of the new system. The methodology integrates the processes of standard database forward design, database reverse engineering and database federation. The discussion is illustrated by a small case study.

1 Introduction

Most large organizations maintain their data in many distinct autonomous databases that have been developed at different times, on different platforms and DMS (Data Management Systems), and most often in independent organizations that have recently merged. The new organizational trends induce them to develop new functions and therefore to make evolve their information system. In most cases, the existing (legacy) functions must be considered and integrated, including the supporting databases. Hence, the need for interoperation frameworks that offer a virtual and integrated view of both the new and legacy functions. We are faced with two distinct engineering domains:

- the classical database forward engineering: designing a global schema that models the new requirements;
- the database federation engineering: developing a database federation that offers an integrated view of the underlying legacy databases (e.g., [7], [5]).

Referring to [3], we consider both of these engineering approaches. In this paper, we propose to combine the forward and federation engineering with the central idea that they are tightly bound. In this way, we state that the global schema is defined not only by the actual requirements but also includes the contribution of the database federation. In this paper, we describe a general architecture based on a conceptual data description and a methodology intended to find out which part of the actual requirements can be covered by the legacy systems and which part has to be managed by additional data sys-

1 The InterDB Project is supported by the Belgian *Région Wallonne*.

tems.

The paper is organized as follows. Section 2 develops a typical case study that allows us to identify some of the main problems to be solved. Section 3 presents the main aspects of the architecture based on a hierarchy data description. Section 4 proposes a general methodology that integrates the forward and federation engineering processes in order to build the components of this architecture in a rigorous way. Section 5 concludes the paper.

2 Motivation

In this section, we develop a small example that illustrates some of the problems we intend to address in this paper. We consider a company in which two manufacturing sites M1 and M2 are active. We also consider the personnel departments P1 and P2 that ensure the HRM of each of these sites, and the sales department S, common to both. Due to historical reasons, the personnel and sales functions of the company are controlled by three independent databases, namely DB-P1 (personnel of site M1), DB-P2 (personnel of site M2) and DB-S (sales of sites M1 and M2). Though the databases are independent, the management applications involve data exchange through asynchronous text files transfer. From a technical point of view, database DB-P1 is made up of a collection of standard COBOL files, while DB-P2 was developed in Oracle V5[2]. DB-S was recently (re)developed with a modern version of IBM DB2.

The new organizational trends force the company to reconsider the structure and objectives of its information system. First, additional functions must be developed to meet new requirements, notably in customer management. Secondly, the existing functions must be integrated, so that the supporting databases are required to be integrated too.

The scenario according to which a quite new system encompassing all the functions of personnel, sales and customer management must be discarded due to too high organizational and financial costs. In particular, the legacy databases cannot be replaced by a unique system, nor even can be reengineered. The company decides to build a virtual database comprising (1) excerpts from the databases DB-P1, DB-P2, DB-S and (2) a new database DB-C that is to support the customer management department, and that will be developed with the object-relational technology. This new architecture will allow new applications to be developed against a unique, consistent, integrated database. It is also decided that some local legacy applications are preserved. This objective raises several critical problems that pertain to two distinct engineering realms, namely *federated databases* and pure *database development*. A new problem also appears: how to distribute the general requirements and the responsibilities of the whole system among the legacy and new components? It is decided to address one integration problem at a time as follows (Fig. 1).

- First, each personnel database is provided with a specific wrapper that yields a semantically rich abstract view of its contents according to a common model (Wrapper P1, Wrapper P2). In particular, these wrappers make explicit, and manage,

2 This version of Oracle ignored the concepts of primary and foreign keys.

hidden constructs and constraints such as foreign keys, that are unknown in the COBOL and Oracle V5 models. Similarly, a wrapper is developed for the DB-S database according to this abstract model (Wrapper S). The main problem in building these wrappers is to recover the exact conceptual schemas of the legacy databases (LCS-P1, LCS-P2, LCS-S) from their physical schemas (LPS-P1, LPS-P2, LPS-S) through reverse engineering techniques. It must be noted that only the data structures that are useful for the future integrated system are made available through these wrappers, in such a way that only export schemas [7] will be considered instead of the complete conceptual schema.

- Then, a common mediator is built on top of these wrappers to reconcile both personnel databases. This component is in charge of integrating the data from both databases by solving data format conflicts, semantic conflicts and data duplication conflicts (the employees that work on both sites are represented in both databases). This unique personnel database is known through its federated conceptual schema FCS-P.

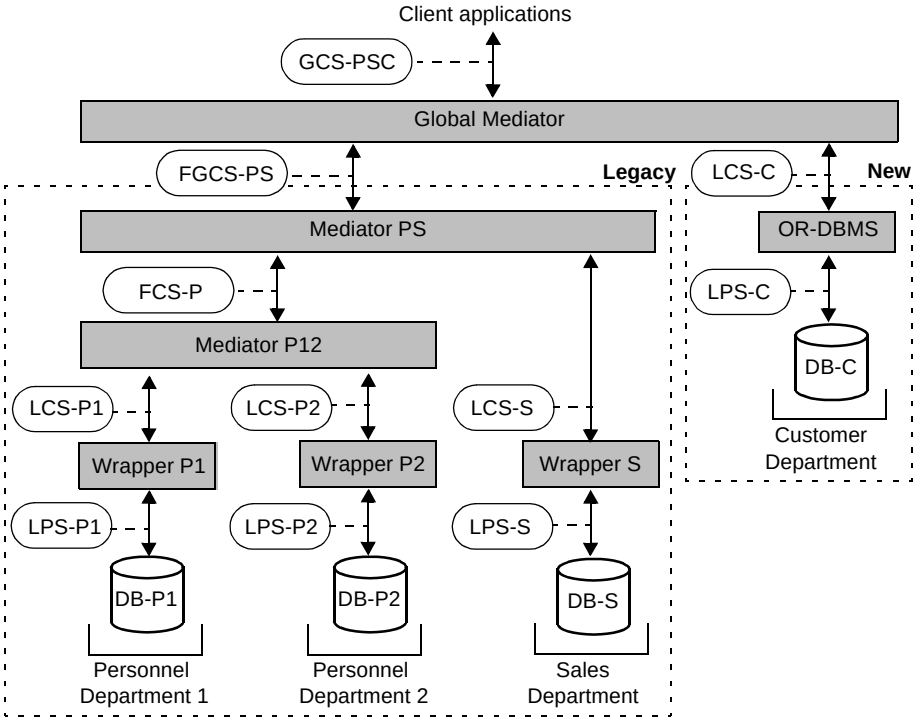

Fig. 1. The new system, available as a unique integrated database with schema GCS-PSC, will comprise a federated database that abstracts and integrates three independent legacy database (COBOL files DB-PP1, Oracle 5 DB-P2, DB2 DB-S), and the new object-relational database DB-C.

- All the databases of the current system are unified under a common mediator that manages the semantic links between the (abstract) personnel database and the sales database. This component completes the structure of the federated database built on

the three legacy databases DB-P1, DB-P2 and DB-S. A common federated global conceptual schema is available, namely FGCS-PS.

- By comparing the services supplied by the federated database against the requirements of the whole system the company wants to develop, and expressed through its global conceptual schema GCS-PSC, the minimum requirements of the new components are elicited. From the corresponding conceptual schema LCS-C, a new object-relational database DB-C is developed, with physical schema LPS-C.
- Finally, in order to provide new applications with a unique database, a global mediator is built, that integrates the federated and the new databases, and that offers a straightforward materialization of the conceptual schema GCS-PSC. Whether the new database is accessed through a wrapper or not, depends on the distance between its data model and the abstract model provided by the mediators. In this example, the Mediator PS model and the DBMS model both are Object-relational. Therefore, the new database need not to be wrapped.

3 Architecture

The architecture comprises a hierarchy of *federated components* (wrappers and mediators) and *new components* (global mediator, new DBMS) (Fig. 2). These components provide a global view that integrates the new needs in information and the existing legacy information.

3.1 Hierarchy Architecture

The architecture defines three classes of schemas, namely, the global schema, the federated schemas and the new schemas. The global schema (GCS) meets the current global information needs by integrating the schemas of the other classes.

The federated schemas comprise the schemas hierarchy that describes the local existing databases. According to the general framework and according to the legacy nature of the databases, each local database source is described by its own Local Physical Schema (LPS) from which a semantically rich description called Local Conceptual Schema (LCS), is obtained through a database reverse engineering process. From this conceptual view, a subset called Local Export Schema (LES) is extracted; it expresses the exact contribution (no more, no less) of this database to the global requirements. The export schemas are merged into a hierarchy of Federated Conceptual Schemas (FCS) (a FCS can be the result of the integration of LES and/or other FCS). The top of this hierarchy is made up of the Federated Global Conceptual Schema (FGCS).

Finally, the new schemas describe the new database through its Local Conceptual Schema (LCS) and its Local Physical Schema (LPS). This database provides the additional required services that cannot be taken in charge by the legacy components.

It is important to note that this architecture does not preclude a wrapped legacy database to serve for more than one mediator, nor to belong to several federated databases. All the conceptual schemas are expressed in a Canonical Data Model which is independent of the underlying technologies. On the other hand, all the physical schemas are based on the data models of their underlying Data Management System.

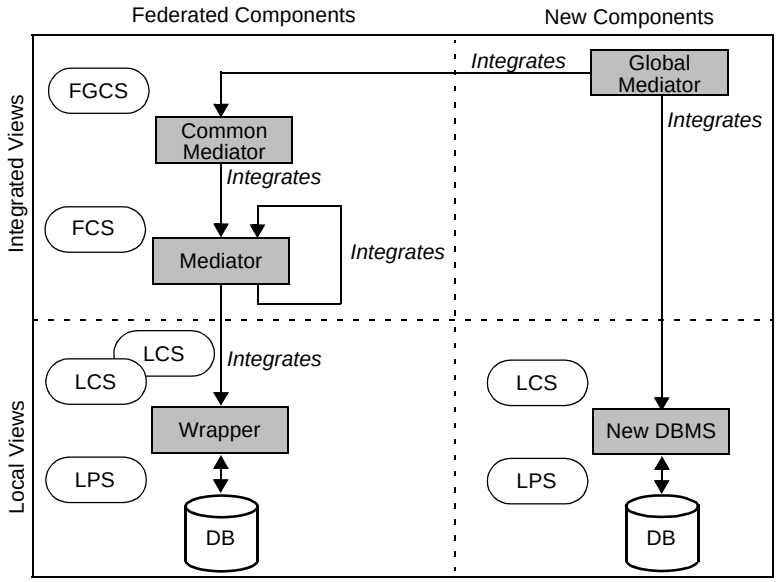

Fig. 2. Abstract model of the architecture.

3.2 Component Architecture

Federated Components. The federated components provide integrated information of existing databases, without the need to physically integrate them [9].

Each *wrapper* controls a legacy database. This software component performs the translation between the LES and the LPS of its database. It offers a abstract interface that hides the specific aspects of a data model family as well as the technical and optimization-dependent constructs of the local database. In addition, it emulates the management of constructs and constraints that exists within the database though they have not been explicitly declared. Foreign keys among COBOL files are examples of such implicit constructs.

A *mediator* offers an abstract integrated view of several legacy databases. One of its main roles is to reconcile independent data structures to yield a unique, coherent, view of the data. In particular, it is to solve format and semantic conflicts, as well as to arbitrate between mutually inconsistent data. In the example of Section 2, one of the mediators builds a consistent description of each personnel, be he/she represented in one database or in both. It operates within a particular domain and uses the services of the wrappers and/or other mediators registered within this domain. Functionally, it offers a conceptual interface based on the FCS that integrates a set of LCS and/or other FCS. It hides details about the location and representation of legacy data. A particular mediator, called the common mediator, is built on top of the hierarchy of wrappers and mediators. It offers the global view of all the components that form the federated database.

New Components. The new components provide global information that are required by the new information system, but is not available in the legacy databases. The global mediator offers an interface based on the GCS that holds all the required information. It

integrates the federated and new databases.

4 Methodology

4.1 Principles

The Global Conceptual Schema (GCS) is the answer to the requirements of the organization as they are perceived from now on. As far as information is concerned, it represents the services that the new system will be able to provide. Since the approach is based on reusing existing resources as far as possible, the future system will comprise legacy databases as well a new one, so that the requirements have to be met by both kinds of databases. Therefore, one of the challenging problems is the precise distribution of the requirements among these components.

We propose to resolve the integration by combining forward and backward processes. Unlike [8], we believe that reverse and forward processes are tightly bound.

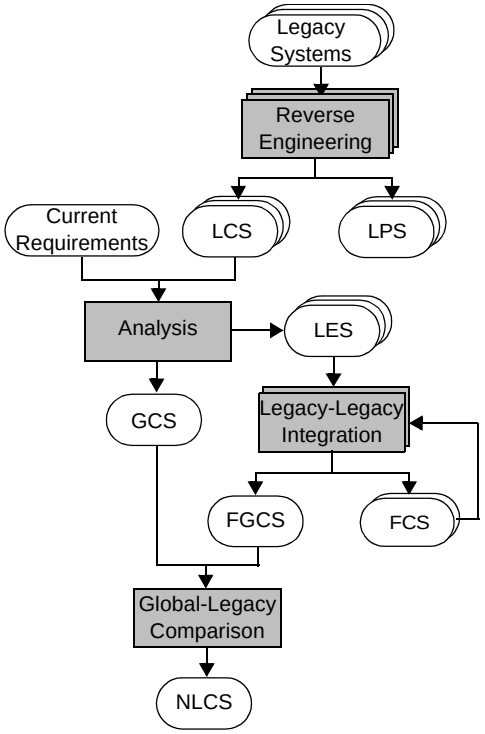

Fig. 3. The main processes of the methodology. The main product is the Global Conceptual Schema (GCS) that integrates the Federated Global Conceptual Schema (FGCS) and the New Local Conceptual Schema (NLCS) that describes the new database. Note that mapping definitions are ignored for simplicity.

The general architecture of the methodology is outlined in Fig. 3. The main processes are: (1) reverse-engineering; (2) analysis; (3) legacy-legacy integration; (4) global-legacy comparison. During these processes, the mappings between the schemas are defined

through schema transformations. In [4] and [5], we have developed a formal transformational approach that is built on a unique extended object-oriented model from which several abstract submodels can be derived by specialization. This approach is intended to provide an elegant way to unify the multiple schemas and mapping descriptions. In the following sections, we will describe briefly the processes of the methodology and the mappings they intend to define. We also illustrate how these processes deal with the hybrid system described in Section 2, the common case study used throughout this paper.

4.2 Reverse-Engineering

This is the process of extracting the Local Physical Schema (LPS) and the Local Conceptual Schema (LCS) of a legacy system. It consists also in defining the corresponding mappings for each legacy database (Fig. 4).

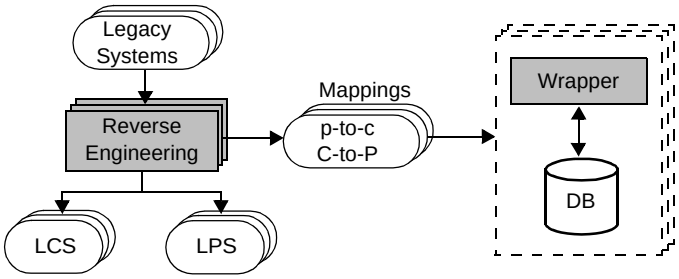

Fig. 4. Reverse engineering of the legacy databases.

4.2.1 Conceptual Schemas Extraction

Since we assume that the databases are legacy systems, we focus on their semantics recovery. Extracting a semantically rich description from a data source is the main goal of the data-centered reverse engineering process (DBRE). The general DBRE methodology we base our approach on has been described in, e.g., [4], [5]). During this process, Local Physical Schemas (LPS) that correspond to the legacy databases are translated into Local Conceptual Schemas (LCS) using the entity-object relationship model. However, we do not assume the quality and the completeness of the physical schemas. In that way, we detect undeclared constructs and constraints in order to provide a semantically rich description of the legacy databases. The reader will find in [5] a more detailed description of this process, which rely heavily on the DBRE approach.

4.2.2 Defining the Mappings

The wrapper is based on the mappings between LPS and LCS. The mappings are modeled through transformations carried out during the reverse engineering process [4]. We assume that deriving a schema from another is performed through techniques such as renaming, translating, making implicit constructs explicit which basically are schema transformations. By processing the process history, mainly made up of traces of transformations, it is possible to derive functional mappings that explain how each conceptual construct is expressed into physical constructs. A wrapper implements these

mappings in two ways: firstly it translates conceptual queries into physical access plans, secondly, it builds conceptual data from physical data extracted from the database.

4.2.3 Application to the Case Study

By analyzing the COBOL programs of DB-P1 and the SQL DDL scripts of DB-P2 and DB-S we can extract the Local Physical Schema (LPS) of each legacy database. Fig. 5 shows the extracted schemas according to their data model.

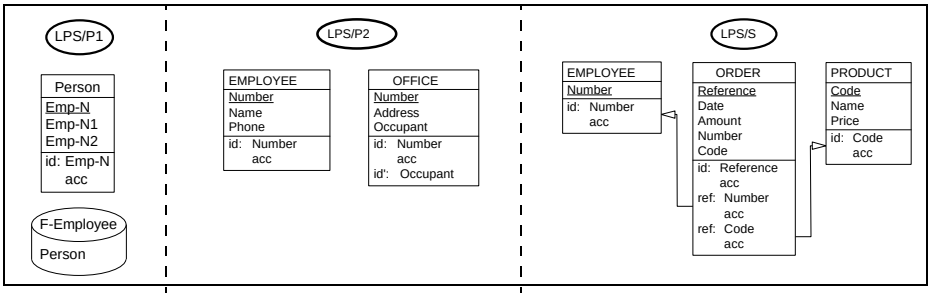

Fig. 5. The local physical schemas. LPS/P1 is made of one file (F-EMPLOYEE) and one record type (PERSON). EMP-N is a unique record key. LPS/P2 comprises the tables EMPLOYEE and OFFICE. LPS/S comprises three tables EMPLOYEE, ORDER and PRODUCT. Keys are represented by the id (primary) and id' (secondary) constructs, foreign keys by the ref construct + a directed arc, and indexes by the acc(ess key) construct.

The LPS extraction is fairly straightforward. However, it recovers explicit constructs only, ignoring all the implicit structures and constraints that have been buried in, e.g., the procedural code of the programs. Hence the need for a refinement process that cleans the physical schema and enriches it with implicit constraints elicited by such techniques as program analysis and data analysis.

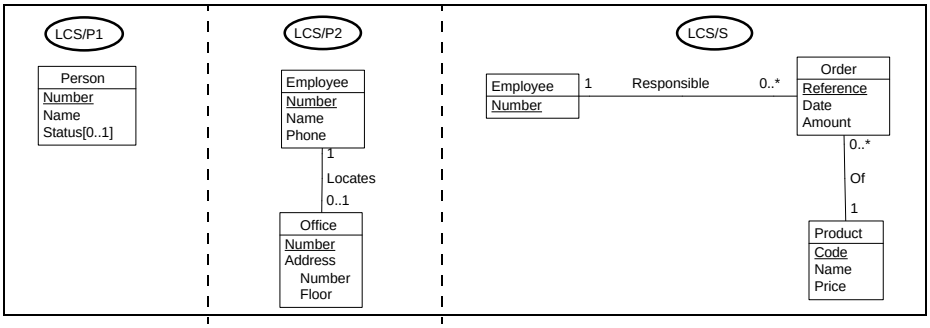

Fig. 6. The local conceptual schemas. In LCS/P, we observe the renaming and the elicitation of an implicit optional attribute. In LCS/P2, a hidden foreign key and an implicit compound attribute have been discovered. In LCS/P2 and LCS/S, the relational foreign keys have been transformed into associations.

The schemas are refined through in-depth inspection of the data, and of the way in which the COBOL program and SQL DML use and manage the data in order to detect the record structures declared in the program sources. For instance, the Oracle V5 DB-P2 database includes a hidden foreign key that can be discovered by looking for, e.g.,

join-based queries. Moreover, names have been made more meaningful and physical constructs are discarded. Finally, the schemas are translated into a same Canonical Data Model (CDM). We therefore obtain the three LCS of Fig. 6. They express the data structures in CDM, enriched by semantic constraints.

4.3 Analysis

The purpose of this process is to define: (1) the Global Conceptual Schema (GCS) that captures the complete requirements of an organization for the future; (2) the Local Export Schemas (LES) that contain the relevant legacy information (Fig. 7). An important characteristics of the approach is the role of the legacy databases in the requirements analysis process. Indeed, since the existing system meets, at least partially, the current needs of the organization, it is an invaluable source of information about the requirements of the future system. On the other hand, the analysis phase can easily identify the objects of the local conceptual schemas that can be reused in the future system. These objects are collected into so-called export schemas.

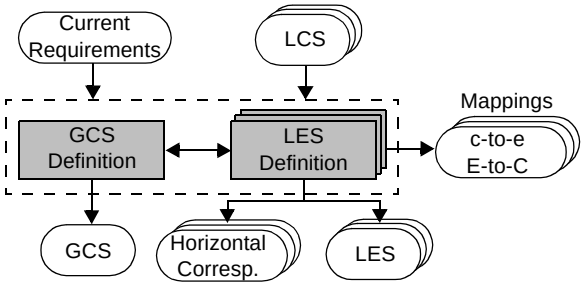

Fig. 7. The analysis process is made up of two subprocesses: the Global Conceptual Schema (GCS) definition and the Local Export Schemas (LES) definition.

4.3.1 Global Conceptual Schema Definition

Collecting and formalizing the requirements of the organization into a conceptual schema is a standard, though still complex, process that has been widely described in the literature. As an implementation of a part of the requirements that are being elaborated, the semantic description of the legacy databases, i.e., their conceptual schemas (LCS) can contribute to the analysis process. The importance of this source of information has been acknowledged and developed in [1] and [2] for instance.

4.3.2 Local Export Schemas Definition

The analysis process identifies and extracts from each local conceptual schema the subset that is still pertinent for the future information system. Each subset a view of the local conceptual schema that is called a Local Export Schema (LES). Since several conceptual schemas are compared, the derived export schema can be restructured in order to reduce their distance.

In addition, this sub-process identifies the horizontal correspondences between potentially overlapping Local Conceptual Schemas, by detecting the similar constructs and their semantic relation. It is based on the knowledge gained by the analyst during

the previous steps of the methodology, on similarities between related parts on the LCS (such as name and structural closeness) and on the instance analysis. This information will drive the building of mediators.

4.3.3 Application to the Case Study

Horizontal Correspondence Definition. The LCS are analyzed in order to detect the horizontal correspondences between them. It appears that the PERSON and EMPLOYEE entity types describe the same application domain population. Data analysis shows that some employees work for the two sites and are represented in both personnel databases. Moreover, data analysis shows that all the employees represented in these databases are recorded in the sales database.

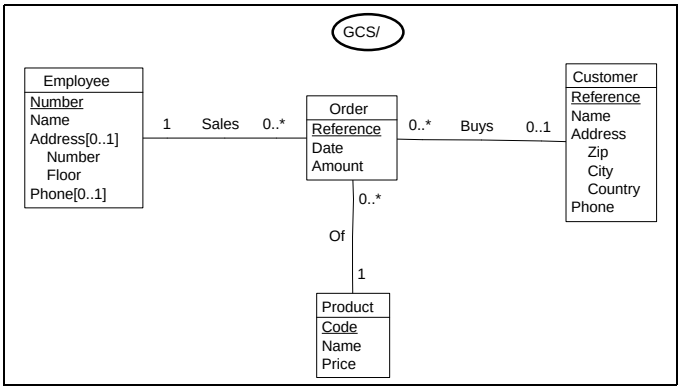

Fig. 8. The global conceptual schema that formalizes the requirements to be met by the future system.

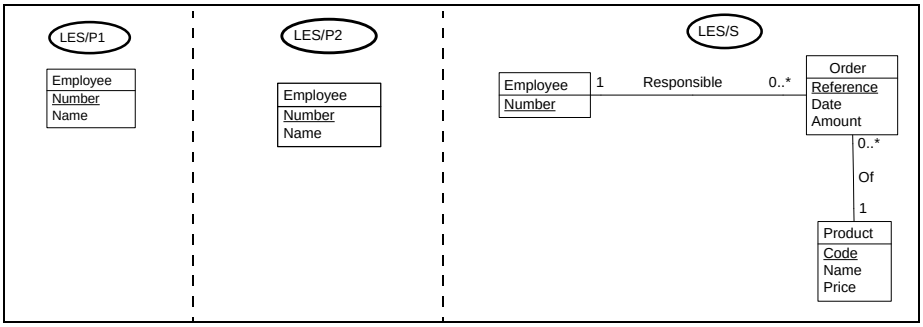

Fig. 9. The local extracted schemas. In LES/P1, the entity type PERSON is renamed EMPLOYEE (distance reduction) and only the attributes NUMBER and NAME are kept. LES/P2 comprises only the entity type EMPLOYEE and its attributes NUMBER and NAME. LES/S contains all the structures of LCS/S.

Global Conceptual Schema Definition. The GCS is defined from the new requirements of the information system and the information held by the personnel databases and sales database. In this way, the GCS integrates legacy information and new information about customers (Fig. 8).

Local Export Schema Definition. By removing from the each LCS the structures and constraints that are not required for the new system, we derive the three Local Export Schemas (LES) (Fig. 9).

4.4 Legacy-Legacy Integration

The objective of this process is to build the hierarchy of integrated schemas and of mediators that is to form the federated database (Fig. 10). In particular, it produces the Federated Global Conceptual Schema (FGCS).

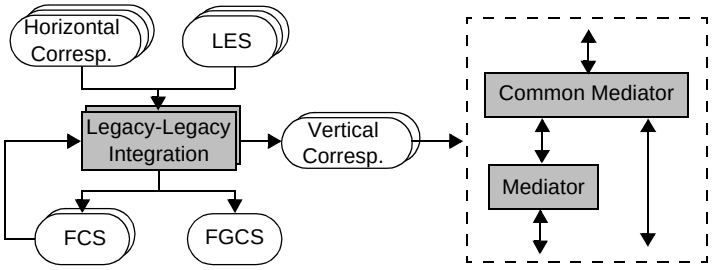

Fig. 10. Legacy-legacy integration.

4.4.1 Principles

Based on the horizontal correspondences defined in the analysis phase, legacy-legacy integration is the process of identifying and solving the conflicts of the LES, and finally, merging them. Identifying similar constructs and merging them is quite easy thanks to the results of the previous steps. Indeed, the reverse engineering step has given the analysts a strong knowledge of the semantics of each construct of the LCS. In addition, the local export schema definition has produced the horizontal correspondences and a view of the legacy databases in which only the relevant information has been kept. Therefore, identifying similar constructs and merging them is much easier than when one processes still unidentified logical schemas as proposed in most federated schema building methodologies.

The results of this process are (1) the Federated Global Conceptual Schema (FGCS) (2) the set of intermediate Conceptual Federated Shemas (FCS); and (3) the vertical correspondences (i.e., the correspondences between the LES and the FCS). The vertical correspondences are formalized as a chain of transformations. As it was shown in [5], most conflicts can be solved through four main transformation techniques: renaming, generalizing, restructuring and discarding.

Defining the intermediate nodes in the hierarchy is an important issue that has yet to be worked out. More specifically, the question is, when two (or more) databases have to be integrated into a common conceptual schema, and managed by the same mediator? Though we have no stable answer to this question yet, we can identify four major dimensions and criteria according to which legacy databases can be grouped and integrated:

- *Similarity of organization objectives.* Both databases focus on the same part of the application domain, so that merging them mimics the merging of these similar organization units.
- *Complementarity of organizational objectives.* The databases control two different parts of the application domains that share some objects, or such that objects from one of them are related with objects from the other one. Merging these databases parallels the merging of two complementary organizational units.
- *Locality.* The databases are located on the same site or on the same server. Merging them produces a virtual database that is identified by its location.
- *Technology.* Two databases are under the control of a unique DBMS. Merging them can be done by a technically homogeneous mediator.

4.4.2 Application to the Case Study

From the horizontal correspondences defined in the previous step, the integration process is fairly easy. We define two federated schemas as follows (Fig. 11).

- The databases DB-P1 and DB-P2 describe the same application domain (HRM). Therefore, they meet the Similarity criterion, so that we suggest to define a virtual Personnel database encompassing both department needs, with schema FCS/P.
- This database has some relationship with the sales database through the concept of Employee. We propose to merge the virtual Personnel database with the sales database. The common schema is also the global description of the federation. It is named FGCS/PS.

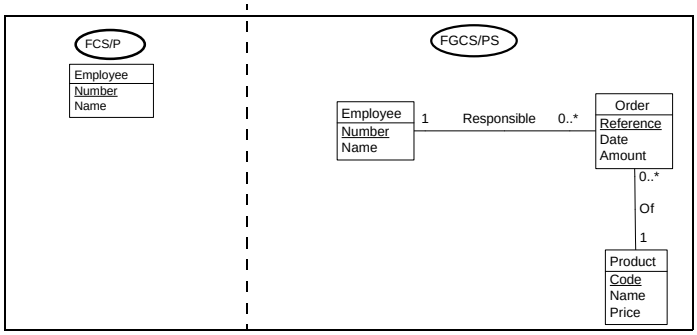

Fig. 11. The conceptual federated schemas. FCS/P integrates the LES of the personnel databases whereas FGCS integrates FCS/P and the LES of the sales department.

4.5 Global-Legacy Comparison

4.5.1 Principles

The global-legacy comparison consists in comparing the GCS that holds the information required by the whole system against the FGCS that integrates the existing information (Fig. 12). Intuitively, the conceptual schema of the new database is obtained by substracting the concepts of the Federated Global Conceptual Schema (FGCS) from the Global Conceptual Schema (GCS). Since the FGCS integrates the LES defined in the same process (analysis) than the GCS, the FGCS is likely to be a subset of GCS, which

simplifies the derivation in many cases. The resulting schema is called the new Local Conceptual Schema (NLCS). It holds the information that is in GCS but not in FGCS.

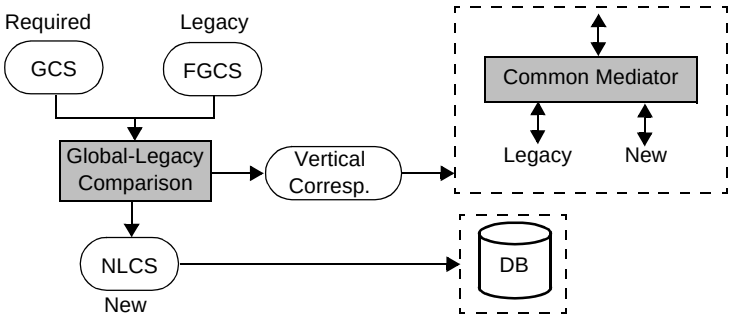

Fig. 12. Global-legacy comparison. This process produces the global mediator and the starting point for the development of the new database.

The vertical correspondences define the mappings between the GCS on the one hand, and the FGCS and NLCS on the other hand. Building an operational database from this conceptual schema is a standard problem [1]. However, some specific issues must be addressed, related with the fact that this new database has to operate in collaboration with the federated database. For instance, some of the important entity types that appear in the federated database may also appear in the new database, either as duplicates (e.g., to represent additional attributes of an entity type, or to improve performances), or as extension (e.g., the personnel of a third department has to be incorporated) of the former ones. The usual architectural issues and problems of distributed databases generally will appear and have to be solved.

4.5.2 Application to the Case Study

By comparing the GCS/PSC against FGCS/PC, we define NLCS/S, the customer schema that holds all the new constructs and a construct (ORDER.REFERENCE) common to the FGCS/PC (Fig. 13). Here, the connection with the federated database is made possible through a kind of foreign key.

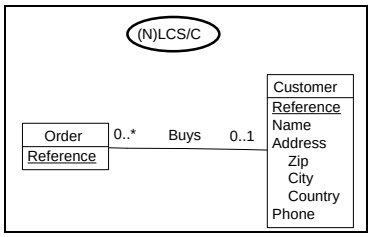

Fig. 13. The local conceptual schema of the new database.

5 Conclusions

One of the biggest challenges of the development of future information systems cer-

tainly is the reuse of valuable legacy components, and more particularly legacy databases. This paper proposed a hierarchical architecture of wrappers and mediators that should bring some important advantages, such as:

- *scalability:* new levels and new databases can be incorporated with minimal cost;
- *progressivity:* each step provides abstract functionalities that can be exploited immediately; this also ensures better development cost and time control, as well as better risk management;
- *legacy preservation:* local databases and applications can be used while their functions meets the local needs;
- *maximal reuse:* since the services (a.o., the semantic structures) of the legacy databases have been precisely elicited, the conceptual schema of new databases to develop includes the new requirements, and only them.

In addition, the paper proposes a systematic methodology to build this architecture. This methodology is based on reverse engineering and transformational techniques to recover the semantics of the legacy databases and to identify the horizontal and vertical mappings between the actual and virtual databases. These mappings allows the generation of wrappers and mediators to be automated [5]. Identifying the minimum contents of the new components is also an objective of the methodology. Nevertheless, some problems remain largely unsolved. One of them is determining the optimal structure of the mediator hierarchy. The paper suggests four criteria that still are to be evaluated in practice.

6 References

1. C. Batini, S. Ceri, , S.B. Navathe, "Conceptual Database Design", Benjamin/Cummings, 1992.
2. M. Blaha, W. Premerlani, "Object-Oriented Modeling and Design for Database Applications", Prentice Hall, 1998.
3. S. Busse, R-D. Kutsche, U. Leser, "Strategies for the Conceptual Design of Federated Information Systems", in Proceedings of EFIS'00, pp. 23-32, IOS Press and Infix, 2000.
4. J-L. Hainaut, J. Henrard, J-M. Hick, D. Roland, V. Englebert, "Database Design Recovery", in Proc. of CAISE'96, Springer-Verlag, 1996.
5. J-L. Hainaut, Ph. Thiran, J-M. Hick, S. Bodart, A. Deflorenne, "Methodology and CASE tools for the development of federated databases", the International Journal of Cooperative Information Systems, Volume 8(2-3), pp. 169-194, World Scientific, June and September, 1999.
6. C. Parent and S. Spaccapietra, "Issues and Approaches of Database Integration", Communications of the ACM, 41(5), pp.166-178, 1998.
7. A.P. Sheth and J.A. Larson "Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases", ACM Computing Surveys, 22(3):183-236, September 1990.
8. W.J. van den Heuvel, W. Hasselbring, M. Papazoglou, "Top-Down Enterprise Application Integration with Reference Models", in Proceedings of EFIS'00, pp. 11-22, IOS Press and Infix, 2000.
9. G. Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer, pp. 38-49, March 1992.
10. Ph. Thiran, J-L. Hainaut, J-M. Hick, A. Chougrani, "Generation of Conceptual Wrappers for Legacy Databases", in Proceedings of DEXA'99, LCNS, Springer-Verlag, September 1999.

Evolving Hybrid Distributed Databases: Architecture and Methodology

Philippe Thiran and Jean-Luc Hainaut

InterDB Project[1], Database Applications Engineering Laboratory
Institut d'Informatique, University of Namur, Belgium
{pth|jlh}@info.fundp.ac.be

Abstract. This paper focuses on the interoperability of autonomous legacy databases with the idea of meeting the future requirements of an organization. It describes a general architecture and a methodology intended to address the problem of providing new client applications with an abstract interface defined from the new requirements and the description of the legacy databases. The architecture relies on a hierarchy of mediators that provide the abstract interface of the new system. The methodology integrates the processes of standard database forward design, database reverse engineering and database federation. The discussion is illustrated by a small case study.

1 Introduction

Most large organizations maintain their data in many distinct autonomous databases that have been developed at different times, on different platforms and DMS (Data Management Systems), and most often in independent organizations that have recently merged. The new organizational trends induce them to develop new functions and therefore to make evolve their information system. In most cases, the existing (legacy) functions must be considered and integrated, including the supporting databases. Hence, the need for interoperation frameworks that offer a virtual and integrated view of both the new and legacy functions. We are faced with two distinct engineering domains:

- the classical database forward engineering: designing a global schema that models the new requirements;
- the database federation engineering: developing a database federation that offers an integrated view of the underlying legacy databases (e.g., [7], [5]).

Referring to [3], we consider both of these engineering approaches. In this paper, we propose to combine the forward and federation engineering with the central idea that they are tightly bound. In this way, we state that the global schema is defined not only by the actual requirements but also includes the contribution of the database federation. In this paper, we describe a general architecture based on a conceptual data description and a methodology intended to find out which part of the actual requirements can be covered by the legacy systems and which part has to be managed by additional data sys-

1 The InterDB Project is supported by the Belgian *Région Wallonne*.

tems.

The paper is organized as follows. Section 2 develops a typical case study that allows us to identify some of the main problems to be solved. Section 3 presents the main aspects of the architecture based on a hierarchy data description. Section 4 proposes a general methodology that integrates the forward and federation engineering processes in order to build the components of this architecture in a rigorous way. Section 5 concludes the paper.

2 Motivation

In this section, we develop a small example that illustrates some of the problems we intend to address in this paper. We consider a company in which two manufacturing sites M1 and M2 are active. We also consider the personnel departments P1 and P2 that ensure the HRM of each of these sites, and the sales department S, common to both. Due to historical reasons, the personnel and sales functions of the company are controlled by three independent databases, namely DB-P1 (personnel of site M1), DB-P2 (personnel of site M2) and DB-S (sales of sites M1 and M2). Though the databases are independent, the management applications involve data exchange through asynchronous text files transfer. From a technical point of view, database DB-P1 is made up of a collection of standard COBOL files, while DB-P2 was developed in Oracle V5[2]. DB-S was recently (re)developed with a modern version of IBM DB2.

The new organizational trends force the company to reconsider the structure and objectives of its information system. First, additional functions must be developed to meet new requirements, notably in customer management. Secondly, the existing functions must be integrated, so that the supporting databases are required to be integrated too.

The scenario according to which a quite new system encompassing all the functions of personnel, sales and customer management must be discarded due to too high organizational and financial costs. In particular, the legacy databases cannot be replaced by a unique system, nor even can be reengineered. The company decides to build a virtual database comprising (1) excerpts from the databases DB-P1, DB-P2, DB-S and (2) a new database DB-C that is to support the customer management department, and that will be developed with the object-relational technology. This new architecture will allow new applications to be developed against a unique, consistent, integrated database. It is also decided that some local legacy applications are preserved. This objective raises several critical problems that pertain to two distinct engineering realms, namely *federated databases* and pure *database development*. A new problem also appears: how to distribute the general requirements and the responsibilities of the whole system among the legacy and new components? It is decided to address one integration problem at a time as follows (Fig. 1).

- First, each personnel database is provided with a specific wrapper that yields a semantically rich abstract view of its contents according to a common model (Wrapper P1, Wrapper P2). In particular, these wrappers make explicit, and manage,

2 This version of Oracle ignored the concepts of primary and foreign keys.

hidden constructs and constraints such as foreign keys, that are unknown in the COBOL and Oracle V5 models. Similarly, a wrapper is developed for the DB-S database according to this abstract model (Wrapper S). The main problem in building these wrappers is to recover the exact conceptual schemas of the legacy databases (LCS-P1, LCS-P2, LCS-S) from their physical schemas (LPS-P1, LPS-P2, LPS-S) through reverse engineering techniques. It must be noted that only the data structures that are useful for the future integrated system are made available through these wrappers, in such a way that only export schemas [7] will be considered instead of the complete conceptual schema.

- Then, a common mediator is built on top of these wrappers to reconcile both personnel databases. This component is in charge of integrating the data from both databases by solving data format conflicts, semantic conflicts and data duplication conflicts (the employees that work on both sites are represented in both databases). This unique personnel database is known through its federated conceptual schema FCS-P.

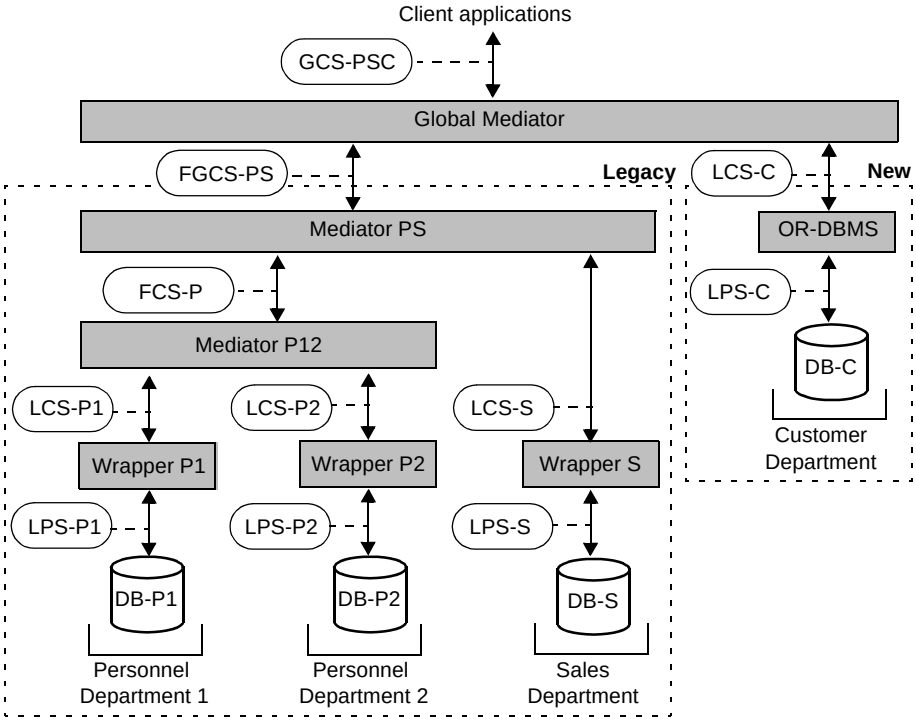

Fig. 1. The new system, available as a unique integrated database with schema GCS-PSC, will comprise a federated database that abstracts and integrates three independent legacy database (COBOL files DB-PP1, Oracle 5 DB-P2, DB2 DB-S), and the new object-relational database DB-C.

- All the databases of the current system are unified under a common mediator that manages the semantic links between the (abstract) personnel database and the sales database. This component completes the structure of the federated database built on

the three legacy databases DB-P1, DB-P2 and DB-S. A common federated global conceptual schema is available, namely FGCS-PS.

- By comparing the services supplied by the federated database against the requirements of the whole system the company wants to develop, and expressed through its global conceptual schema GCS-PSC, the minimum requirements of the new components are elicited. From the corresponding conceptual schema LCS-C, a new object-relational database DB-C is developed, with physical schema LPS-C.
- Finally, in order to provide new applications with a unique database, a global mediator is built, that integrates the federated and the new databases, and that offers a straightforward materialization of the conceptual schema GCS-PSC. Whether the new database is accessed through a wrapper or not, depends on the distance between its data model and the abstract model provided by the mediators. In this example, the Mediator PS model and the DBMS model both are Object-relational. Therefore, the new database need not to be wrapped.

3 Architecture

The architecture comprises a hierarchy of *federated components* (wrappers and mediators) and *new components* (global mediator, new DBMS) (Fig. 2). These components provide a global view that integrates the new needs in information and the existing legacy information.

3.1 Hierarchy Architecture

The architecture defines three classes of schemas, namely, the global schema, the federated schemas and the new schemas. The global schema (GCS) meets the current global information needs by integrating the schemas of the other classes.

The federated schemas comprise the schemas hierarchy that describes the local existing databases. According to the general framework and according to the legacy nature of the databases, each local database source is described by its own Local Physical Schema (LPS) from which a semantically rich description called Local Conceptual Schema (LCS), is obtained through a database reverse engineering process. From this conceptual view, a subset called Local Export Schema (LES) is extracted; it expresses the exact contribution (no more, no less) of this database to the global requirements. The export schemas are merged into a hierarchy of Federated Conceptual Schemas (FCS) (a FCS can be the result of the integration of LES and/or other FCS). The top of this hierarchy is made up of the Federated Global Conceptual Schema (FGCS).

Finally, the new schemas describe the new database through its Local Conceptual Schema (LCS) and its Local Physical Schema (LPS). This database provides the additional required services that cannot be taken in charge by the legacy components.

It is important to note that this architecture does not preclude a wrapped legacy database to serve for more than one mediator, nor to belong to several federated databases. All the conceptual schemas are expressed in a Canonical Data Model which is independent of the underlying technologies. On the other hand, all the physical schemas are based on the data models of their underlying Data Management System.

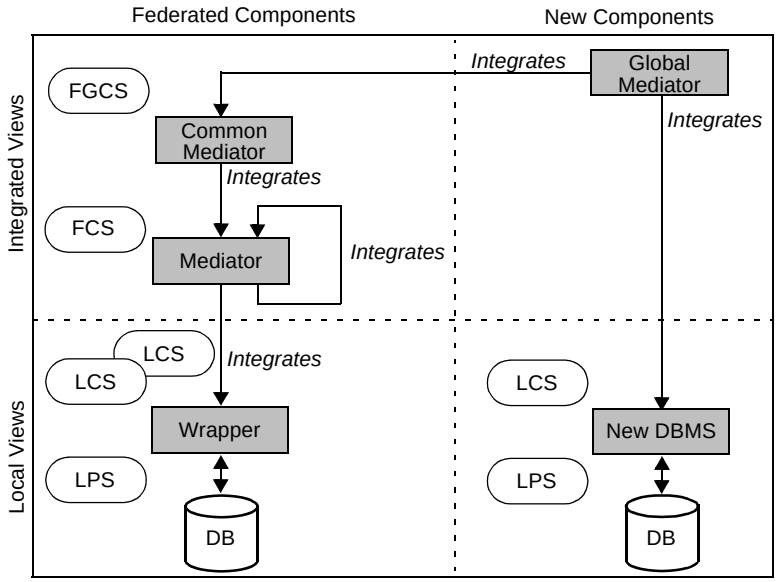

Fig. 2. Abstract model of the architecture.

3.2 Component Architecture

Federated Components. The federated components provide integrated information of existing databases, without the need to physically integrate them [9].

Each *wrapper* controls a legacy database. This software component performs the translation between the LES and the LPS of its database. It offers a abstract interface that hides the specific aspects of a data model family as well as the technical and optimization-dependent constructs of the local database. In addition, it emulates the management of constructs and constraints that exists within the database though they have not been explicitly declared. Foreign keys among COBOL files are examples of such implicit constructs.

A *mediator* offers an abstract integrated view of several legacy databases. One of its main roles is to reconcile independent data structures to yield a unique, coherent, view of the data. In particular, it is to solve format and semantic conflicts, as well as to arbitrate between mutually inconsistent data. In the example of Section 2, one of the mediators builds a consistent description of each personnel, be he/she represented in one database or in both. It operates within a particular domain and uses the services of the wrappers and/or other mediators registered within this domain. Functionally, it offers a conceptual interface based on the FCS that integrates a set of LCS and/or other FCS. It hides details about the location and representation of legacy data. A particular mediator, called the common mediator, is built on top of the hierarchy of wrappers and mediators. It offers the global view of all the components that form the federated database.

New Components. The new components provide global information that are required by the new information system, but is not available in the legacy databases. The global mediator offers an interface based on the GCS that holds all the required information. It

integrates the federated and new databases.

4 Methodology

4.1 Principles

The Global Conceptual Schema (GCS) is the answer to the requirements of the organization as they are perceived from now on. As far as information is concerned, it represents the services that the new system will be able to provide. Since the approach is based on reusing existing resources as far as possible, the future system will comprise legacy databases as well a new one, so that the requirements have to be met by both kinds of databases. Therefore, one of the challenging problems is the precise distribution of the requirements among these components.

We propose to resolve the integration by combining forward and backward processes. Unlike [8], we believe that reverse and forward processes are tightly bound.

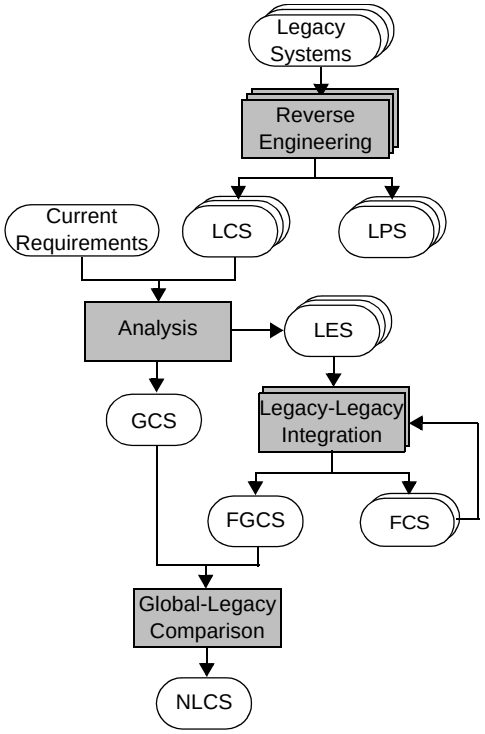

Fig. 3. The main processes of the methodology. The main product is the Global Conceptual Schema (GCS) that integrates the Federated Global Conceptual Schema (FGCS) and the New Local Conceptual Schema (NLCS) that describes the new database. Note that mapping definitions are ignored for simplicity.

The general architecture of the methodology is outlined in Fig. 3. The main processes are: (1) reverse-engineering; (2) analysis; (3) legacy-legacy integration; (4) global-legacy comparison. During these processes, the mappings between the schemas are defined

through schema transformations. In [4] and [5], we have developed a formal transformational approach that is built on a unique extended object-oriented model from which several abstract submodels can be derived by specialization. This approach is intended to provide an elegant way to unify the multiple schemas and mapping descriptions. In the following sections, we will describe briefly the processes of the methodology and the mappings they intend to define. We also illustrate how these processes deal with the hybrid system described in Section 2, the common case study used throughout this paper.

4.2 Reverse-Engineering

This is the process of extracting the Local Physical Schema (LPS) and the Local Conceptual Schema (LCS) of a legacy system. It consists also in defining the corresponding mappings for each legacy database (Fig. 4).

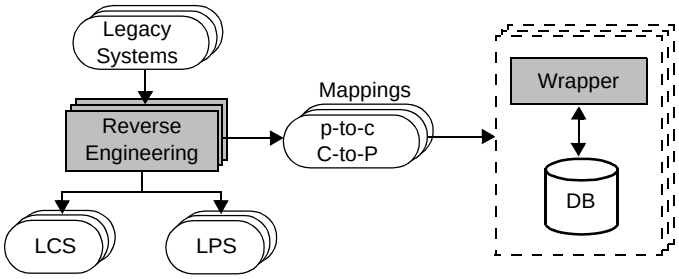

Fig. 4. Reverse engineering of the legacy databases.

4.2.1 Conceptual Schemas Extraction

Since we assume that the databases are legacy systems, we focus on their semantics recovery. Extracting a semantically rich description from a data source is the main goal of the data-centered reverse engineering process (DBRE). The general DBRE methodology we base our approach on has been described in, e.g., [4], [5]). During this process, Local Physical Schemas (LPS) that correspond to the legacy databases are translated into Local Conceptual Schemas (LCS) using the entity-object relationship model. However, we do not assume the quality and the completeness of the physical schemas. In that way, we detect undeclared constructs and constraints in order to provide a semantically rich description of the legacy databases. The reader will find in [5] a more detailed description of this process, which rely heavily on the DBRE approach.

4.2.2 Defining the Mappings

The wrapper is based on the mappings between LPS and LCS. The mappings are modeled through transformations carried out during the reverse engineering process [4]. We assume that deriving a schema from another is performed through techniques such as renaming, translating, making implicit constructs explicit which basically are schema transformations. By processing the process history, mainly made up of traces of transformations, it is possible to derive functional mappings that explain how each conceptual construct is expressed into physical constructs. A wrapper implements these

mappings in two ways: firstly it translates conceptual queries into physical access plans, secondly, it builds conceptual data from physical data extracted from the database.

4.2.3 Application to the Case Study

By analyzing the COBOL programs of DB-P1 and the SQL DDL scripts of DB-P2 and DB-S we can extract the Local Physical Schema (LPS) of each legacy database. Fig. 5 shows the extracted schemas according to their data model.

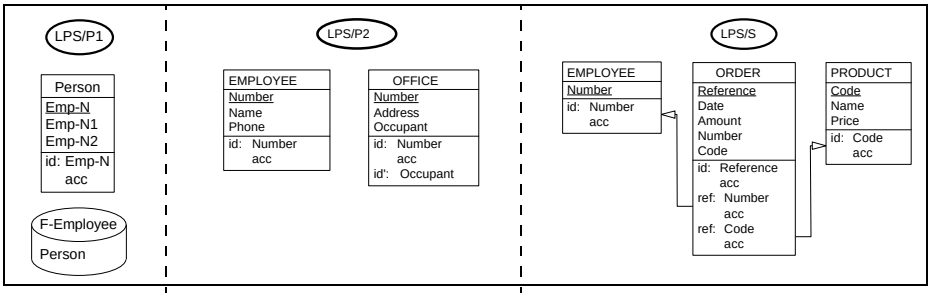

Fig. 5. The local physical schemas. LPS/P1 is made of one file (F-EMPLOYEE) and one record type (PERSON). EMP-N is a unique record key. LPS/P2 comprises the tables EMPLOYEE and OFFICE. LPS/S comprises three tables EMPLOYEE, ORDER and PRODUCT. Keys are represented by the id (primary) and id' (secondary) constructs, foreign keys by the ref construct + a directed arc, and indexes by the acc(ess key) construct.

The LPS extraction is fairly straightforward. However, it recovers explicit constructs only, ignoring all the implicit structures and constraints that have been buried in, e.g., the procedural code of the programs. Hence the need for a refinement process that cleans the physical schema and enriches it with implicit constraints elicited by such techniques as program analysis and data analysis.

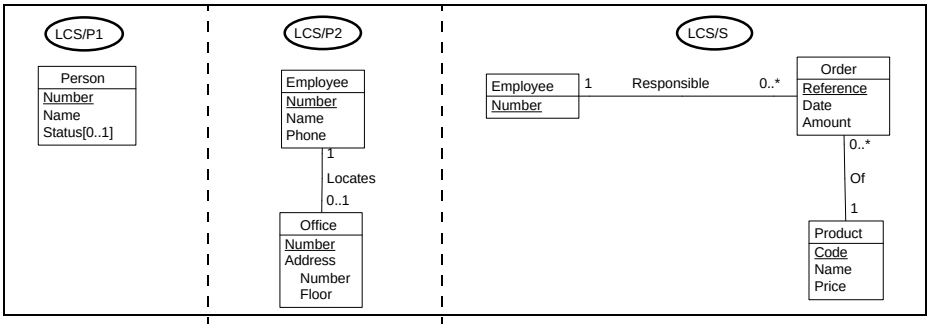

Fig. 6. The local conceptual schemas. In LCS/P, we observe the renaming and the elicitation of an implicit optional attribute. In LCS/P2, a hidden foreign key and an implicit compound attribute have been discovered. In LCS/P2 and LCS/S, the relational foreign keys have been transformed into associations.

The schemas are refined through in-depth inspection of the data, and of the way in which the COBOL program and SQL DML use and manage the data in order to detect the record structures declared in the program sources. For instance, the Oracle V5 DB-P2 database includes a hidden foreign key that can be discovered by looking for, e.g.,

join-based queries. Moreover, names have been made more meaningful and physical constructs are discarded. Finally, the schemas are translated into a same Canonical Data Model (CDM). We therefore obtain the three LCS of Fig. 6. They express the data structures in CDM, enriched by semantic constraints.

4.3 Analysis

The purpose of this process is to define: (1) the Global Conceptual Schema (GCS) that captures the complete requirements of an organization for the future; (2) the Local Export Schemas (LES) that contain the relevant legacy information (Fig. 7). An important characteristics of the approach is the role of the legacy databases in the requirements analysis process. Indeed, since the existing system meets, at least partially, the current needs of the organization, it is an invaluable source of information about the requirements of the future system. On the other hand, the analysis phase can easily identify the objects of the local conceptual schemas that can be reused in the future system. These objects are collected into so-called export schemas.

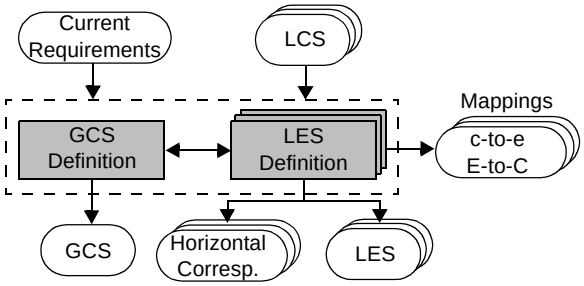

Fig. 7. The analysis process is made up of two subprocesses: the Global Conceptual Schema (GCS) definition and the Local Export Schemas (LES) definition.

4.3.1 Global Conceptual Schema Definition

Collecting and formalizing the requirements of the organization into a conceptual schema is a standard, though still complex, process that has been widely described in the literature. As an implementation of a part of the requirements that are being elaborated, the semantic description of the legacy databases, i.e., their conceptual schemas (LCS) can contribute to the analysis process. The importance of this source of information has been acknowledged and developed in [1] and [2] for instance.

4.3.2 Local Export Schemas Definition

The analysis process identifies and extracts from each local conceptual schema the subset that is still pertinent for the future information system. Each subset a view of the local conceptual schema that is called a Local Export Schema (LES). Since several conceptual schemas are compared, the derived export schema can be restructured in order to reduce their distance.

In addition, this sub-process identifies the horizontal correspondences between potentially overlapping Local Conceptual Schemas, by detecting the similar constructs and their semantic relation. It is based on the knowledge gained by the analyst during

the previous steps of the methodology, on similarities between related parts on the LCS (such as name and structural closeness) and on the instance analysis. This information will drive the building of mediators.

4.3.3 Application to the Case Study

Horizontal Correspondence Definition. The LCS are analyzed in order to detect the horizontal correspondences between them. It appears that the PERSON and EMPLOYEE entity types describe the same application domain population. Data analysis shows that some employees work for the two sites and are represented in both personnel databases. Moreover, data analysis shows that all the employees represented in these databases are recorded in the sales database.

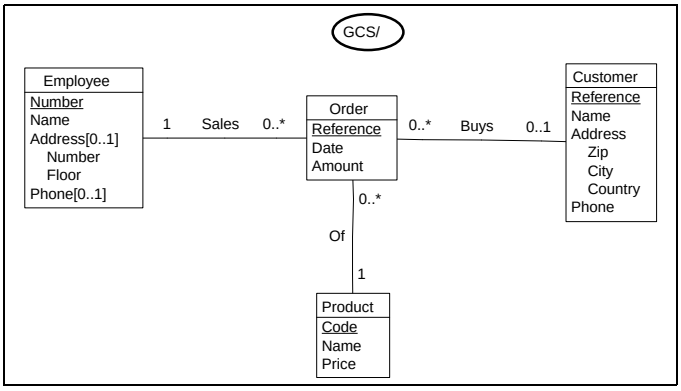

Fig. 8. The global conceptual schema that formalizes the requirements to be met by the future system.

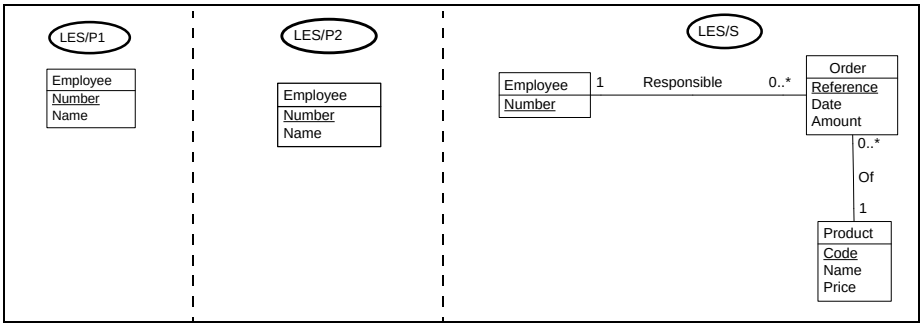

Fig. 9. The local extracted schemas. In LES/P1, the entity type PERSON is renamed EMPLOYEE (distance reduction) and only the attributes NUMBER and NAME are kept. LES/P2 comprises only the entity type EMPLOYEE and its attributes NUMBER and NAME. LES/S contains all the structures of LCS/S.

Global Conceptual Schema Definition. The GCS is defined from the new requirements of the information system and the information held by the personnel databases and sales database. In this way, the GCS integrates legacy information and new information about customers (Fig. 8).

Local Export Schema Definition. By removing from the each LCS the structures and constraints that are not required for the new system, we derive the three Local Export Schemas (LES) (Fig. 9).

4.4 Legacy-Legacy Integration

The objective of this process is to build the hierarchy of integrated schemas and of mediators that is to form the federated database (Fig. 10). In particular, it produces the Federated Global Conceptual Schema (FGCS).

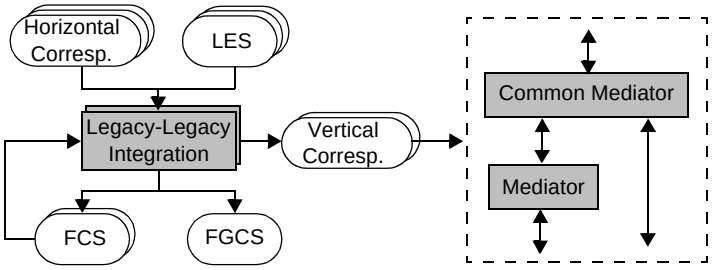

Fig. 10. Legacy-legacy integration.

4.4.1 Principles

Based on the horizontal correspondences defined in the analysis phase, legacy-legacy integration is the process of identifying and solving the conflicts of the LES, and finally, merging them. Identifying similar constructs and merging them is quite easy thanks to the results of the previous steps. Indeed, the reverse engineering step has given the analysts a strong knowledge of the semantics of each construct of the LCS. In addition, the local export schema definition has produced the horizontal correspondences and a view of the legacy databases in which only the relevant information has been kept. Therefore, identifying similar constructs and merging them is much easier than when one processes still unidentified logical schemas as proposed in most federated schema building methodologies.

The results of this process are (1) the Federated Global Conceptual Schema (FGCS) (2) the set of intermediate Conceptual Federated Shemas (FCS); and (3) the vertical correspondences (i.e., the correspondences between the LES and the FCS). The vertical correspondences are formalized as a chain of transformations. As it was shown in [5], most conflicts can be solved through four main transformation techniques: renaming, generalizing, restructuring and discarding.

Defining the intermediate nodes in the hierarchy is an important issue that has yet to be worked out. More specifically, the question is, when two (or more) databases have to be integrated into a common conceptual schema, and managed by the same mediator? Though we have no stable answer to this question yet, we can identify four major dimensions and criteria according to which legacy databases can be grouped and integrated:

- *Similarity of organization objectives.* Both databases focus on the same part of the application domain, so that merging them mimics the merging of these similar organization units.
- *Complementarity of organizational objectives.* The databases control two different parts of the application domains that share some objects, or such that objects from one of them are related with objects from the other one. Merging these databases parallels the merging of two complementary organizational units.
- *Locality.* The databases are located on the same site or on the same server. Merging them produces a virtual database that is identified by its location.
- *Technology.* Two databases are under the control of a unique DBMS. Merging them can be done by a technically homogeneous mediator.

4.4.2 Application to the Case Study

From the horizontal correspondences defined in the previous step, the integration process is fairly easy. We define two federated schemas as follows (Fig. 11).

- The databases DB-P1 and DB-P2 describe the same application domain (HRM). Therefore, they meet the Similarity criterion, so that we suggest to define a virtual Personnel database encompassing both department needs, with schema FCS/P.
- This database has some relationship with the sales database through the concept of Employee. We propose to merge the virtual Personnel database with the sales database. The common schema is also the global description of the federation. It is named FGCS/PS.

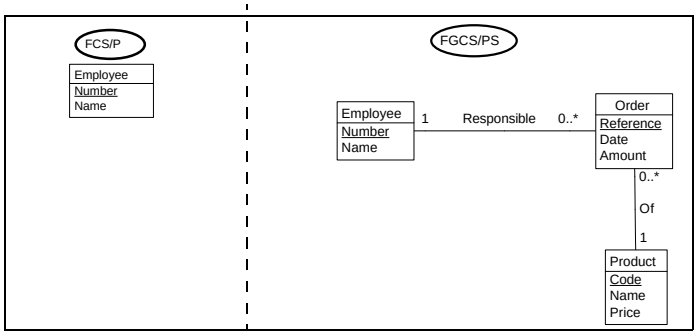

Fig. 11. The conceptual federated schemas. FCS/P integrates the LES of the personnel databases whereas FGCS integrates FCS/P and the LES of the sales department.

4.5 Global-Legacy Comparison

4.5.1 Principles

The global-legacy comparison consists in comparing the GCS that holds the information required by the whole system against the FGCS that integrates the existing information (Fig. 12). Intuitively, the conceptual schema of the new database is obtained by substracting the concepts of the Federated Global Conceptual Schema (FGCS) from the Global Conceptual Schema (GCS). Since the FGCS integrates the LES defined in the same process (analysis) than the GCS, the FGCS is likely to be a subset of GCS, which

simplifies the derivation in many cases. The resulting schema is called the new Local Conceptual Schema (NLCS). It holds the information that is in GCS but not in FGCS.

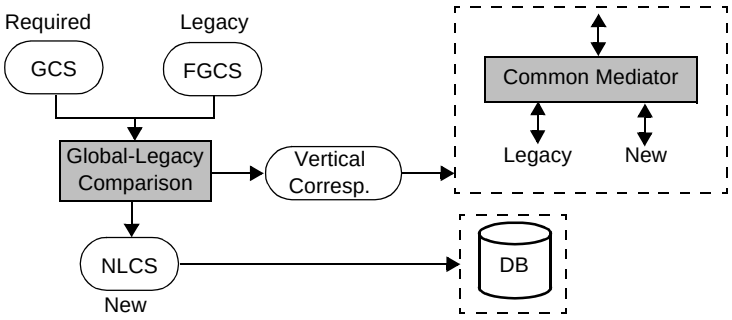

Fig. 12. Global-legacy comparison. This process produces the global mediator and the starting point for the development of the new database.

The vertical correspondences define the mappings between the GCS on the one hand, and the FGCS and NLCS on the other hand. Building an operational database from this conceptual schema is a standard problem [1]. However, some specific issues must be addressed, related with the fact that this new database has to operate in collaboration with the federated database. For instance, some of the important entity types that appear in the federated database may also appear in the new database, either as duplicates (e.g., to represent additional attributes of an entity type, or to improve performances), or as extension (e.g., the personnel of a third department has to be incorporated) of the former ones. The usual architectural issues and problems of distributed databases generally will appear and have to be solved.

4.5.2 Application to the Case Study

By comparing the GCS/PSC against FGCS/PC, we define NLCS/S, the customer schema that holds all the new constructs and a construct (ORDER.REFERENCE) common to the FGCS/PC (Fig. 13). Here, the connection with the federated database is made possible through a kind of foreign key.

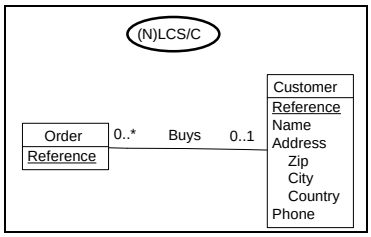

Fig. 13. The local conceptual schema of the new database.

5 Conclusions

One of the biggest challenges of the development of future information systems cer-

tainly is the reuse of valuable legacy components, and more particularly legacy databases. This paper proposed a hierarchical architecture of wrappers and mediators that should bring some important advantages, such as:

- *scalability:* new levels and new databases can be incorporated with minimal cost;
- *progressivity:* each step provides abstract functionalities that can be exploited immediately; this also ensures better development cost and time control, as well as better risk management;
- *legacy preservation:* local databases and applications can be used while their functions meets the local needs;
- *maximal reuse:* since the services (a.o., the semantic structures) of the legacy databases have been precisely elicited, the conceptual schema of new databases to develop includes the new requirements, and only them.

In addition, the paper proposes a systematic methodology to build this architecture. This methodology is based on reverse engineering and transformational techniques to recover the semantics of the legacy databases and to identify the horizontal and vertical mappings between the actual and virtual databases. These mappings allows the generation of wrappers and mediators to be automated [5]. Identifying the minimum contents of the new components is also an objective of the methodology. Nevertheless, some problems remain largely unsolved. One of them is determining the optimal structure of the mediator hierarchy. The paper suggests four criteria that still are to be evaluated in practice.

6 References

1. C. Batini, S. Ceri, , S.B. Navathe, "Conceptual Database Design", Benjamin/Cummings, 1992.
2. M. Blaha, W. Premerlani, "Object-Oriented Modeling and Design for Database Applications", Prentice Hall, 1998.
3. S. Busse, R-D. Kutsche, U. Leser, "Strategies for the Conceptual Design of Federated Information Systems", in Proceedings of EFIS'00, pp. 23-32, IOS Press and Infix, 2000.
4. J-L. Hainaut, J. Henrard, J-M. Hick, D. Roland, V. Englebert, "Database Design Recovery", in Proc. of CAISE'96, Springer-Verlag, 1996.
5. J-L. Hainaut, Ph. Thiran, J-M. Hick, S. Bodart, A. Deflorenne, "Methodology and CASE tools for the development of federated databases", the International Journal of Cooperative Information Systems, Volume 8(2-3), pp. 169-194, World Scientific, June and September, 1999.
6. C. Parent and S. Spaccapietra, "Issues and Approaches of Database Integration", Communications of the ACM, 41(5), pp.166-178, 1998.
7. A.P. Sheth and J.A. Larson "Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases", ACM Computing Surveys, 22(3):183-236, September 1990.
8. W.J. van den Heuvel, W. Hasselbring, M. Papazoglou, "Top-Down Enterprise Application Integration with Reference Models", in Proceedings of EFIS'00, pp. 11-22, IOS Press and Infix, 2000.
9. G. Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer, pp. 38-49, March 1992.
10. Ph. Thiran, J-L. Hainaut, J-M. Hick, A. Chougrani, "Generation of Conceptual Wrappers for Legacy Databases", in Proceedings of DEXA'99, LCNS, Springer-Verlag, September 1999.