HODFA: An Architectural Framework for Homogenizing Heterogeneous Legacy Databases[1]

Kamalakar Karlapalem   Qing Li   Chung-Dak Shum

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
e-mail: {kamal, qing, shum}@cs.ust.hk

Abstract

One of the main difficulties in supporting global applications over a number of localized databases and migrating legacy information systems to modern computing environment is to cope with the heterogeneities of these systems. In this paper, we present a *novel flexible architecture* (called HODFA) to dynamically connect such localized heterogeneous databases in forming a *homogenized federated database system* and to support the process of transforming a collection of heterogeneous information systems onto a homogeneous environment. We further develop an incremental methodology of homogenization in the context of our HODFA framework, which can facilitate different degrees of homogenization in a stepwise manner, so that existing applications will not be affected during the process of homogenization.

1 Introduction and Motivation

In many large organizations a number of autonomous legacy *localized databases* owned by the departments or sections of an organization are maintained and managed to support a set of applications. These local databases tend to be heterogeneous, that is, managed by different database management systems (DBMS) based on different models (like hierarchical, network, relational, object oriented). With the organizational data spread over various parts of the organization there is a need to provide a common platform over these localized databases that support global data processing applications to (i) maintaining consistency of local databases, (ii) generating consolidated global reports, (iii) supporting local and global on-line transaction processing, and (iv) decision support systems.

Information systems (IS) came into existence long before the introduction of any DBMS. These ISs may consist of a large collection of programs and data, usually written in COBOL, and use a legacy database service, for example, IBM's IMS. They are important assets and are critical for the day-to-day operation of an organization. Today, these *legacy information systems* pose one of the most difficult information management problems for many large organizations. The cost of maintaining a legacy IS often takes up a major portion of all the IS resources. Lack of documentation, inflexibility in the design, poor performance, inappropriate functionality all attribute to the high cost. The lack of understanding of a legacy IS also makes it difficult for organizations to take full advantage of newer technologies, such as client-server architectures, and current software, such as relational DBMS. The migration of legacy IS to very flexible modern computing environment is an important undertaking that we will address in this paper.

One of the main difficulties in supporting global applications over a number of localized databases is to cope with the heterogeneities of these systems. These systems were not originally designed to facilitate any cooperation and there is no general model for interoperability among such isolated software systems. Should these systems exist under a homogeneous environment, global applications built upon a common set of tools and services can be developed much more efficiently. Legacy ISs in a homogeneous environment will also be much simpler, easier to understand and more readily to be migrated to better computing environment in future. In this paper, we explore the concept of *homogenization*: the process of transforming a collection of heterogeneous legacy ISs onto a homogeneous environment.

[1] A two page synopsis of this work will appear as a poster paper in 7th International conference on Parallel and Distributed Computing Systems, Las Vegas, 1994.

We propose a novel flexible architecture to dynamically connect these localized heterogeneous databases in forming a *homogenized federated database system* to support the above mentioned applications. This architecture is based on the concept of *homogenization of federated databases* by managing and maintaining a *mirror copy* of each of the localized databases (as a member database) in a single robust DBMS. By interconnecting and homogenizing localized databases, we want to achieve the following specific goals: (1) to provide the ability to streamline the replacement of the legacy localized databases; (2) new global applications at different levels of abstraction and scale can be developed on top of the homogenized federated databases; (3) provide interoperability among a set of heterogeneous databases so that previously isolated heterogeneous localized databases can be loosely coupled and become interoperable.

The rest of the paper is organized as follows. In Section 2, we present an overview of the proposed architecture (HODFA), in Section 3, we elaborate on a practical incremental methodology of conducting the homogenization process within HODFA framework, and examine the issues of supporting migration and development of applications, in Section 4, we compare our approach with others work. We conclude this paper with a summary and future work in Section 5.

2 An Overview of the Architecture

Our approach for homogenization of disparate databases is supported by a flexible architecture shown in Figure 1. This architecture, termed as HODFA (for *HOmogenized Database Federation Architecture*), consists of a hybrid of preexisting and new ISs. We shall first briefly introduce the various components of this architecture by describing their functionality and interrelationships. We will then evaluate this architecture with respect to our goals.

2.1 Components of HODFA

As shown in Figure 1, HODFA is composed of a set of *localized databases* managed by pre-existing local database systems (LDS), a set of local applications accessing the local databases, a set of *member databases* that are "mirror copies" of localized databases, a set of applications (MASS) that manage the data movement between localized and member databases, a *coordinator DBMS* with multilevel materialized view support system, a case base (CB) and a system data/knowledge directory (SD). These are detailed immediately below.

2.1.1 Localized and Member Databases

Localized databases are a set of preexisting databases each of which is managed by a local database management system. These databases support a set of localized applications for the individual department and section use, and are totally autonomous and possibly heterogeneous.

Member databases are derived databases through what we call "mirroring processes" (see below). In particular, each member database is a *mirror copy* of a single localized database. The purpose of the mirroring process is to convert the localized heterogeneous databases into a set of homogeneous databases which can be efficiently managed by a single robust DBMS (namely, the coordinator DBMS). Depending on the requirements of the global applications and the sensitivity of the data in localized databases, a member database (in the coordinator DBMS) may be created through mirroring its corresponding localized database, either as a full copy of data, or partial copy of data, or a view of the data, or only just the schema definition with no data. Note that if all the member databases consist of only schema definitions of localized databases and no data, then we have the classical federated database system architecture. If all the member databases are replicated consistent full copies of localized databases, then we have homogenized the localized (heterogeneous) databases. In all other scenarios we have some degree of homogenization of the localized databases. In general member databases can be populated by the *mirroring application support system* as described below.

2.1.2 Mirroring application support system

In HODFA, the process of homogenizing localized heterogeneous databases into member databases is conducted by tailored (customized) Mirroring Application Support Scheme (MASS) components. Each MASS component consists of a set of applications to extract data from a localized database and populate a member database and vice versa. In case of full or partial homogenization (of localized databases), the MASS components will also be responsible for maintaining the consistency between the localized databases and their corresponding member databases. There can be potentially many MASS modules moving data among localized and member databases and maintaing consistency between them. Depending on the organization needs, the mirroring/homogenizing processes can be conducted in various ways, including (i) to extract the relevant data from localized database into an intermediate medium (like tapes) and populate the member database from this intermediate medium, (ii) to use the localized applications to update both the localized database and its corresponding member database simultaneously, (iii) to employ a classical federated database management system to support maintenance of both localized and member databases, (iv) to populate the member databases from the manual records.

2.1.3 Coordinator DBMS

The coordinator DBMS is a single robust DBMS that supports, manages and maintains the member databases. The coordinator DBMS also provides capabilities like transaction management services, recovery, and access to multiple member databases (interoperability). This is a major advantage

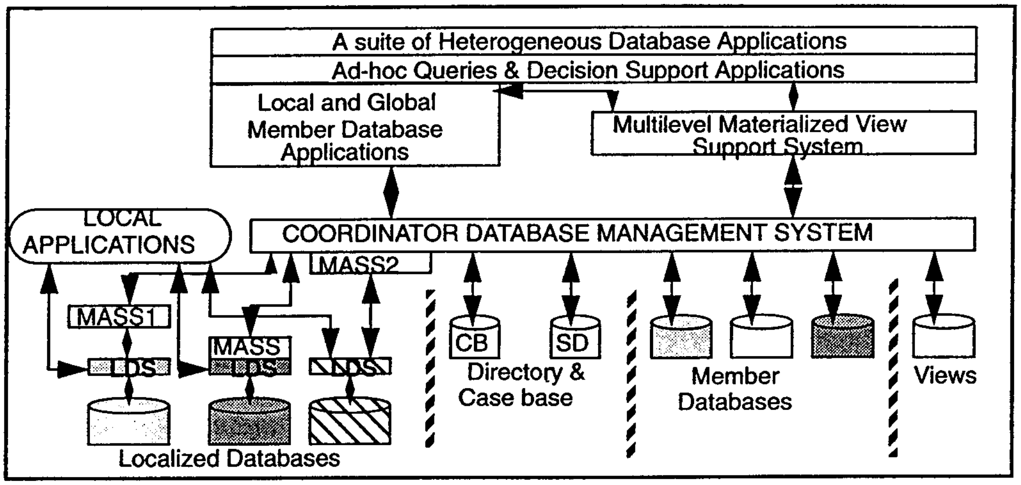
A suite of Heterogeneous Database Applications
Ad-hoc Queries & Decision Support Applications
Local and Global Member Database Applications
Multilevel Materialized View Support System
LOCAL APPLICATIONS
COORDINATOR DATABASE MANAGEMENT SYSTEM
MASS2
MASS1
MASS
LDS
LDS
LDS
CB
SD
Directory & Case base
Member Databases
Views
Localized Databases

Figure1. HODFA - HOmogenized Database Federation Architecture

for having a coordinator DBMS because lack of advanced transaction management services and interoperability has impeded the development of full fledged heterogeneous database system based solution. Since the schemas for all member databases are defined using the same data model (on which coordinator DBMS is based) the problem of schema integration reduces to a much easier problem of resolving semantic conflicts between member database schemas. The coordinator DBMS provides the access to the member databases for the MASS and all the applications including (emulated) localized and newly developed (global) applications. The coordinator DBMS is also responsible for supporting the maintenance of global consistency between the member databases, and identification of the inconsistencies in the localized databases.

2.1.4 Local Applications

Each of the localized databases has a set of applications that access it. These local applications are processed at the sites of the localized databases. Therefore, these local applications have to be migrated on to the coordinator database system so as to support them. Migration of local applications can be achieved by: (1) rewriting the code by changing just the database access statements in the applications (if both localized databases and member databases are based on same data model), (2) designing, implementing and testing new applications from scratch, and (3) emulating the local applications by a combination of 4GLs and user interface.

2.1.5 Miscellaneous Components

There are some other components used by HODFA to support miscellaneous functions, which also interact with the above components. Specifically, there is a system data/knowledge directory (SD) to maintain consolidated information about the processing requirements of applications on the member databases, and to facilitate homogenization. A case base (CB) is used to store the access plans of frequently issued ad-hoc queries or transactions. This can enable the coordinator DBMS to efficiently support these queries and transactions. In case the coordinator DBMS is a distributed database system (see below), then the information in the system directory and case base could be used for distributed database design to efficiently process the applications. Within the coordinator DBMS, a view mechanism is also included to support different kinds of applications with different levels of abstraction. The coordinator DBMS supports the specification and maintenance of materialized views so as to efficiently execute decision support applications and ad-hoc queries.

2.2 Evaluation of the Architecture

We shall now evaluate the proposed architecture with respect to our general objectives listed in Section 1.

Legacy database systems replacement: By using the HODFA approach to homogenize localized heterogeneous databases, it is now possible for any of the outdated/legacy databases to be upgraded (and ported) to more advanced data model (and hardware platform), and be able to eventually get replaced by its corresponding fully homogenized member databases. Obviously it will be more desirable for the existing localized applications to be able to continue function during the upgrade and replacement period. This calls for a practical methodology (see Section 3) for conducting the homogenization of the localized databases properly.

*Support for new global applications:*The architecture set up for HODFA is not only suitable for replacing legacy localized database systems by member databases, it is also flexible and extensible in supporting different kinds of distributed/global applications on top of it. In particular, with a multilevel materialized view support system to be incorporated by the coordinator DBMS (see Figure 1), HODFA allows a wide range of applications to be developed using a view mechanism. With the help of the case base (CB) and the system data/knowledge directory (SD), it is possible for HODFA to derive global/distributed constraints by employing some efficient machine learning techniques, and maintain such constraints through interactions with individual member databases.

Facilitate interoperability: The feasibility of devising the desired level of interoperability between the localized database system and the coordinator DBMS depends on several aspects. Besides the necessary networking facilities (for interconnectivity), other issues need to be resolved include the resolution of system and semantic heterogeneity, the derivation and/or integration of schemas and views, and of course, the successful implementation of the system in terms of its key components and their interactions. On the first two issues, there have been "standard" algorithms and techniques resulted from the past two decades' research [Batini86, Ceri86], which are both efficient and robust, and are readily available so that they can be applied to our system. Once the homogenization of disparate localized databases is completed, the problem of facilitating interoperability becomes simple as all the member databases are managed by a single robust coordinator DBMS.

Feasibility for implementing HODFA: The key components involved are obviously the coordinator DBMS and the MASS mechanism, with the latter mainly relying on schema conversion and program translation techniques. For the coordinator DBMS, there is a choice among some of the robust and powerful DBMSs such as relational and/or object-oriented ones; further, there is also an orthogonal issue of whether to use a centralized or distributed DBMS for the coordinator DBMS. This depends a lot on the scope and the size of the applications to be supported. Fortunately, both centralized and distributed DBMSs supporting either relational or object-oriented models are commercially available now, and most of these systems can be run on both PCs and mainframes.

3 Fundamentals of Homogenization

Given a set of heterogeneous localized databases, there are many aspects involved in transforming them into a homogeneous environment (the process of this transformation is known as *homogenization).*

3.1 A Classification of Homogenization

Intuitively, homogenization is a process that creates and maintains a mirror copy of a localized database as a *member database* in a federated database system. The resultant member databases have a "homogeneous" environment. Homogenization can be accomplished at different degrees. There can be zero-degree homogenization (no-homogenization), that is, only localized database schema is defined in the coordinator database, but the member database does not contain any data. At the other end we can have full homogenization, wherein the member database is a replicated consistent copy of the localized database. In all other cases we have some partial-degree of homogenization (or partial homogenization).

Manner of Homogenization: A mechanism is needed by which the data gets populated into a member database from the corresponding localized database. This can be done in a stepwise manner, or at one-time. Any degree of homogenization can be implemented by appropriately defining the (sub-)schema/view on the localized database as the member database schema, and then possibly populating this member database. This is known as one-time homogenization, and any change in degree of homogenization would require redefining the member database schema. *Stepwise homogenization* consists of gradual homogenization of the localized databases. Clearly, stepwise homogenization takes longer time from the viewpoint of homogenizing legacy database systems, but can be more flexible and desirable from the perspective of supporting existing and new applications simultaneously. In this paper, our focus is on stepwise, partial or full degree, homogenization methodology, which is supported by HODFA as described below.

3.2 HODFA's Incremental Homogenization

The main reason for homogenizing localized databases is to facilitate a "graceful" transition and (eventual) replacement of the legacy systems. In HODFA, an *incremental* homogenization methodology is devised for this purpose, which facilitates the following features and capabilities:

Co-existence of Member and Localized Databases: It is almost impossible to have all the localized databases and their applications to be transformed and ported at one time to member databases under the coordinated DBMS without affecting the applications. It is therefore desirable for the transition to be taken place in a gradual, piecemeal fashion. Thus there shall be populated localized and member databases being accessed by the applications, hence co-existence of these localized and member databases becomes a paramount issue. The data that is common to both member and localized databases needs to be kept consistent. Interoperability among member and localized database must be supported to enable applications to access both localized and member databases. The degree of homogenization can be monitored by two approaches: i) *application based:* move each application and all the data that is relevant to it from localized database to member database, ii) *database object based:* move each object of database (like relation, entity) from the localized database to the member database. The homogenization is complete once all the applications or complete set of database objects have been moved from localized to member databases.

Flexibility of the Mirroring Process: For homogenization to take place, it is the function of MASS -- a middle layer of software between the localized databases (LDs) and member databases (MDs) -- to move the data from LDs to MDs. Ideally, data movement is from localized to member databases only, but because of the fault-tolerant nature of the HODFA it is required to provide data movement from the MDs to LDs as well. As HODFA aims to support homogenization from zero-degree homogenization to full homogenization, MASS has to be a flexible system. For zero-degree homogenization the interoperability between the localized databases and member databases is the main issue, whereas for partial/full homogenization the consistency between localized databases and member databases is the main issue.

Relocation of Localized Applications: A major problem with homogenization of localized databases is the support for the relocation of applications from localized database systems to the coordinator DBMS. There are many ways through which applications can be moved from localized databases systems to coordinator DBMS:

- *Rewrite the applications on top of the coordinator DBMS.* If there are a large number of localized applications that need to be moved then software filters can be developed to facilitate this movement. There are also filters available to transform code from one language to an other language (like from PASCAL to 'C', or FORTRAN to 'C', etc.).
- *Employ software re-engineering techniques.* There are software re-engineering techniques that facilitate program conversion, evaluation of the database requirements of the applications, and detection of inherent constraints encoded in the applications (see [Rugaber90]). These techniques not only facilitate relocation of applications from LDS to coordinator DBMS, but also provide inputs for member database design.
- *Use advanced application program generators to emulate the localized applications.* Most DBMSs provide an application development environment based on CASE that facilitates fast application development and testing. This application development environment can be utilized to implement the localized applications on the coordinator DBMS. One major advantage of this approach is that the consistency criteria (that is, the constraints that need to be satisfied by the member databases) can be specified declaratively as part of the application. Thus any inconsistencies between the consistency criteria among member databases can be detected.

Impact on the Applications: Eventually, when all the localized databases are homogenized to member databases which are managed by a single robust DBMS (i.e., the coordinator DBMS), all the applications become operational on top of the coordinated DBMS. There are two kinds of applications that can be identified here: those that correspond to existing (localized) applications, and those that are newly developed. For the former, continuous, uninterrupted operations are guaranteed by the incremental homogenization approach.

As the localized databases within the same organization are integrated into a homogenized database federation, it also creates a great opportunity for the organization to develop a wide range of *new* applications (both centralized and distributed ones) on top of the federation. Let us consider some specific types of applications HODFA is targeted to support on top of the coordinated DBMS. As depicted in Figure 1, these include applications like decision support systems (DSS) that mainly access the database federation through the multilevel materialized views. The next set of applications are the ad-hoc queries; these queries facilitate the identification of similar data objects between member databases, and they accommodate dynamic schema integration. The access plans for the ad-hoc queries can be stored in a case base and used to increase the efficiency of the ad-hoc queries. The OLTP applications cater to the day to day access and update of the member databases, which call for fast response time and updateable views. Finally we have a set of global and local member database applications that emulate (part of) the localized database applications, or new local member database applications, or new global database applications that access more than one member database.

Besides an extended scope and high-level new applications that can be supported, our approach can also facilitate a powerful feature called "semantic relativism" [McLeod80], meaning that different applications can hold different views/interpretations on the data of the same (federated) system. For example, DSS type of applications may wish to keep semantic relativism in interpreting the data and predicting changes, and emulated local MD applications may need to exercise application autonomy during (and even after) the transition from the homogenization. In HODFA, this will be made possible by its multilevel materialized view mechanism. Further, the homogenized database federation can also maintain so-called "application autonomy" [Sheth90] existed in the previously localized legacy systems; such local autonomy can be quite important and essential to support for not only old applications but also many of the new applications.

4 Related Work

In a federated database system [Heimbigner85-1,Sheth90-1], there is no global schema and no central coordinating authority so that local autonomy and heterogeneity can be maintained by local databases that participate in the federation. Schema integration is a process of generating a global schema by resolving the conflicts between a set of distributed database schemas [Batini86]. There is no emphasis on homogenization of the disparate/heterogeneous distributed databases. HODFA facilitates schema integration as all the schemas are specified using the logical data model provided by the coordinator DBMS. Interoperability is a function of a federated database system which specifies the amount of cooperative database access that is possible between two participating database systems [IMS91, Bukhres93]. Interoperability enables homogenization of federated databases, and homogenization in HODFA implies (to a great extent) interoperability. Interoperability is necessary in HODFA to maintain the consistency between member and localized databases; it also enables applications to access both localized databases and member databases. Neither schema integration nor interoperability facilitate replacement of the localized databases by the member databases. [Brodie93] in their methodology for migrating legacy information systems investigate the general framework for incremental migration, but do not consider the concept of homogenization of heterogeneous localized databases. Since their methodology is very general, it is complicated both from architecture and methodology point of view. Moreover, the maintenance of the database consistency in their approach will be much more complicated.

5 Summary and Future Work

We have presented an architecture that facilitate dynamic evolution of localized heterogeneous database into a homogenized database federation. The approach we are taking is based on the concept of homogenization -- a process of transforming localized databases into homogeneous member databases via mirroring. An incremental methodology of homogenization is proposed in the context of our HODFA framework, which can facilitate different degrees of homogenization in a stepwise manner. Some notable characteristics and possible features of this approach are:

- ability to streamline the replacement of the legacy localized databases by the member database.
- providing a robust coordinator database management system as a front-end to the heterogeneous localized databases.
- providing the support for managing and maintaining a mirror copy of the localized database at logical and/or physical levels as member databases.
- ability to dynamically integrate the member database schemes based on application semantics.

The HODFA system will be implemented by employing several DBMSs on a network of Sun4 workstations. Some of the implementation issues are: development of incremental homogenization of heterogeneous databases, maintaining consistency between localized and member databases, dynamic schema integration to enable efficient processing of applications, and maintenance of member databases, monitoring and management of the homogenized federated database system, and defining multilevel materialized views to cater to the data requirements of decision support systems.

We are currently collaborating with a telecommunication company to migrate their legacy information systems to a homogenized environment using this methodology.

References

[Batini86]C. Batini, M. Lenzerni and S.B. Navathe. A comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4), pp 323-364, 1986.

[Brodie93]M. L. Brodie and M. Stonebraker. DARWIN: On the Incremental Migration of Legacy Information Systems. TR-0222-10-92-165, GTE Laboratories, Inc. March 1993.

[Bukhres93]O.A. Bukhres, J. Chen, W. Du and A.K. Elmagarmid. InterBase: An Execution Environment for Heterogeneous Software Systems. IEEE *Computer*, 26(8), pp 57-69, 1993.

[Ceri84] S. Ceri and G. Pellagatti. *Distributed Databases: Principles and Systems*, McGraw-Hill, New York, 1984.

[Heimbigner85]D. Heimbigner and D. McLeod. A federated database architecture *ACM Trans. on Office Information Systems*, 3(3), pp 253-278, 1985.

[IMS91] Y. Kambayashi, M. Rusinkiewicz and A. Sheth. *Proceedings of the 1st Int'l Workshop on Interoperability in Multidatabase Database Systems*, IEEE Computer Society, 1991.

[McLeod81]D. McLeod and J.M. Smith. Abstractions in Databases. In *Proc. Workshop on Data Abstraction, Databases and Conceptual Modeling*, Pingree Park, CO, June 1980.

[Ozsu91] M. T. Ozsu and P. Valduriez. Principles of Distributed Database Systems. Prentice-Hall publishers, 1991.

[Rothnie77]J.B. Rothnie and N. Goodman A survey of research and development in distributed database management. In *Proceedings of 2nd VLDB Conference*, 1977.

[Rugaber90] S. Rugaber, S. B. Ornburn and R. J. LeBlanc Jr. Recognizing design decisions in programs, IEEE *Software* vol. 7, no. 1 (jan 1990) pp 46-54.

[Sheth90]A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), pp 183-236, 1990.

[Stonebraker77]M.R. Stonebraker and E. Neuhold. A distributed database version of INGRES. In *Proceedings of Berkeley Workshop on Distributed Data Management and Computer Networks*, pp.19-36, 1977.