VHDBS: A Federated Database System for Electronic Commerce

Bernhard Holtkamp
Norbert Weissenberg
Fraunhofer ISST
Postfach 520 130, D-44207 Dortmund
{weissenb, holtkamp}@do.isst.fhg.de

Xuequn Wu
Deutsche Telekom AG
Technologiezentrum
Postfach 100 003, D-64276 Darmstadt
wu@tzd.telekom.de

Abstract

During a German Telekom industrial research project, we designed and developed the VHDBS system, a federated database system for integrating information from pre-existing and legacy databases. We are now going to apply VHDBS for eCommerce scenarios, where database access is of central importance.

VHDBS is based on the OMG/CORBA 2.0 and ODMG 2.0 standards and offers an object-oriented data model described in ODMG/ODL as well as the query language ODMG/OQL. In this paper we discuss possible application domains and aspects of the technical background of VHDBS, especially its architecture, repository concept and Web user interface.

1. Introduction

ECommerce is gaining momentum. While only a few companies have successfully established eCommerce applications in the consumer market, interest in business-to-business applications is growing.

Key functionality for eCommerce is information integration. Prior to the availability of the WWW, federated database systems [LMR90] [TTC+90] [ÖV91] [HBP93] [BE96] [Con97] [ZCF+97] were developed to enable access to distributed information in a heterogeneous environment with partially autonomous sites. Due to missing standards and overall system complexity, the use of federated database systems was limited to a few, highly specialized applications. Only recently, WWW and Java [Eck97], OMG/CORBA and IIOP [OMG96a], ODMG and OQL [ODMG97] have established a basis for building systems with standardized interfaces, and standard components like ORBs and the emerging CORBAservices [OMG96b] are available as building blocks.

On this basis we designed and developed the VHDBS federated database system prototype [WW97] [Wu96], which provides a way to integrate pre-existing as well as legacy database systems in a distributed and heterogeneous environment. With VHDBS, different departments of an enterprise may be integrated forming a virtual Intranet, and different enterprises may be combined forming an Extranet (i.e. a virtual subnet for a closed user group based on Internet technology). The database system architectures and data schemas used may differ vastly, since VHDBS provides flexible integration and mapping mechanisms. All applications and database systems already in use remain unchanged, since VHDBS only provides additional benefits via its database adapters to be installed at each enterprise location.

Various projects have recently been carried out on the integration of heterogeneous database systems (e.g. Dataplex [Chu90], Mariposa [SAL+96], Infinity [HST97], [Kim93] and [Su96]). Most of these projects focus on schema translation technique and on resolving the syntactic and semantic heterogeneities that arise upon integration. These systems show weakness by system scalability. Often, only database systems based on some given data models can be integrated into such systems, and adding a local database system is a costly task. Unlike these systems, VHDBS relies on an extensable object-oriented multi-tier architecture and an object-oriented data model that can be easily mapped to different data models. Furthermore, it provides a new flexible repository concept for specifying integrated views. The approaches presented in [UA95] and [KFA94] employ object-oriented queries and an object-oriented data model for the federated level. However, the local databases are restricted to relational databases. The Garlic project [Car95] supports the integration of different databases similar to VHDBS in many ways. Moreover, Garlic focuses on the integration of sources that are not necessarily structured databases. Unlike VHDBS, Garlic's database servers are "thin" modules that support the directly specified queries. While extensions can be made easily on every level by VHDBS, all extensions have to be made on the federated level by Garlic. IRO-DB [BFH+94] [BFN94], Pegasus [SAD+95] and OpenDM/Efendi [Mid96], however, are very near to VHDBS, but do not use CORBA and have a different repository concept.

The paper is organized as follows: Section 2 describes possible application domains and benefits of business-to-business eCommerce for VHDBS. Section 3 gives the technical background and Section 4 shows how the technique is used to provide the benefits in eCommerce applications for a particular scenario.

2. Application Domains and Benefits of VHDBS

The following two scenarios of using VHDBS illustrate its wide application domain from Intranet to Extranet. Both scenarios can be realized either by using a possibly emerging VHDBS Telekom service (thus outsourcing IT management and maintenance to a service provider) or by installing VHDBS servers within the enterprise itself. Moreover, the scenarios are variable according to the locations at which VHDBS components (export and/ or query components) are accessible.

2.1. Application Domain “Enterprise-wide Information System“

The following scenario illustrates the use of VHDBS as a means for integrating distributed information into an enterprise-wide information system: A public relations agency has branch offices in different locations nationwide or worldwide. Each office runs PR campaigns for a variety of customer companies and products. Corresponding records are kept in local files or databases. To gain a new contract for a particular PR campaign, an office needs a list of references to successful campaigns with similar requirements already performed by the agency. Thus, the office has to check the records of all other offices. Contacting colleagues often results in a time-consuming data collection process with call-backs and follow-up calls. Using VHDBS would provide the desired information automatically within minutes.

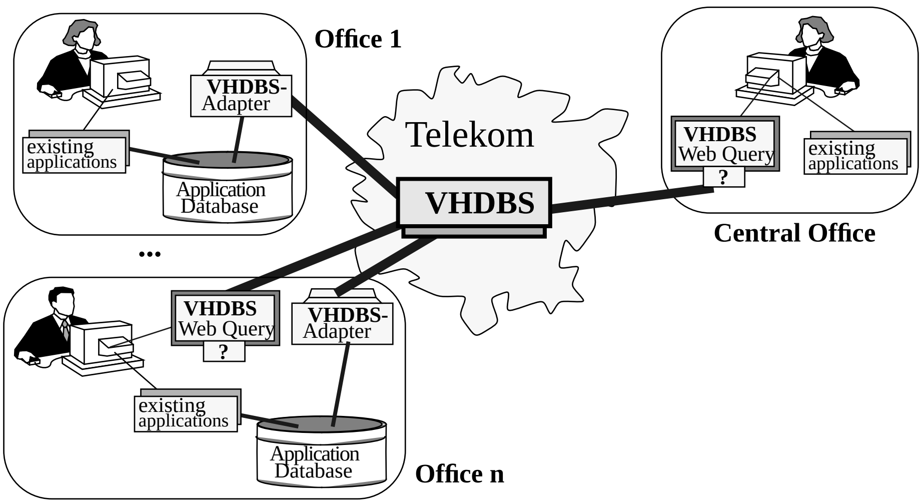

Figure 1: VHDBS as a Basis for an Enterprise-wide Information System

This sample scenario also holds for many other application domains. For example, a personnel service corporation may access application documents and open positions from remote databases in a similar way. It depends on whether only a central department may access all data, or whether each department is allowed to collect enterprise-wide information as depicted in Figure 1. In summary, VHDBS provides the following benefits within this scenario:

- enterprise-wide information and global information becomes available at all locations
- remote queries and queries across different data sources are treated like local queries
- homogeneous local access to heterogeneous distributed data sources becomes possible
- legacy systems may be fully integrated, and legacy data is reused without migration
- autonomous access to plugged-in database systems remains
- business processes and communications are improved

2.2. Application Domain "Integration of Business Partners"

Given an enterprise with a number of suppliers having a partially overlapping range of products or parts, in order to procure a particular part, the enterprise has to query all relevant suppliers and evaluate their offers. Even in business fields with very short reaction times such as the automobile industry, this process takes several hours. Using VHDBS, procurement may be optimized as follows: all suppliers provide VHDBS access to their products and supply databases. Then the enterprise can quickly access and sort all relevant supplier information. The suppliers do not have to change their IT infrastructure, they only need to install a VHDBS database adapter and may still control which information is provided via VHDBS.

Virtual enterprises may benefit even more using VHDBS as a Telekom service: A broker combines offers of his business partners to construct his own product catalogue, which in a second step can be easily imported into an Internet storefront built up by means of existing solutions like Intershop [Int97] or IBM's Net.Commerce [IBM97]. Providers only have to install a VHDBS adapter for their product catalogue and the broker uses a VHDBS interface to compose and store his offers.

A general system architecture for such business-to-business applications is depicted in Figure 2 and this scenario will be detailed in Section 4. The benefits in this case are analogous to those of an enterprise-wide information system, but here the connected locations are highly independent and there is much more heterogeneity. Of course both domains can be combined in scenarios where some business partners also have enterprise-wide information systems.

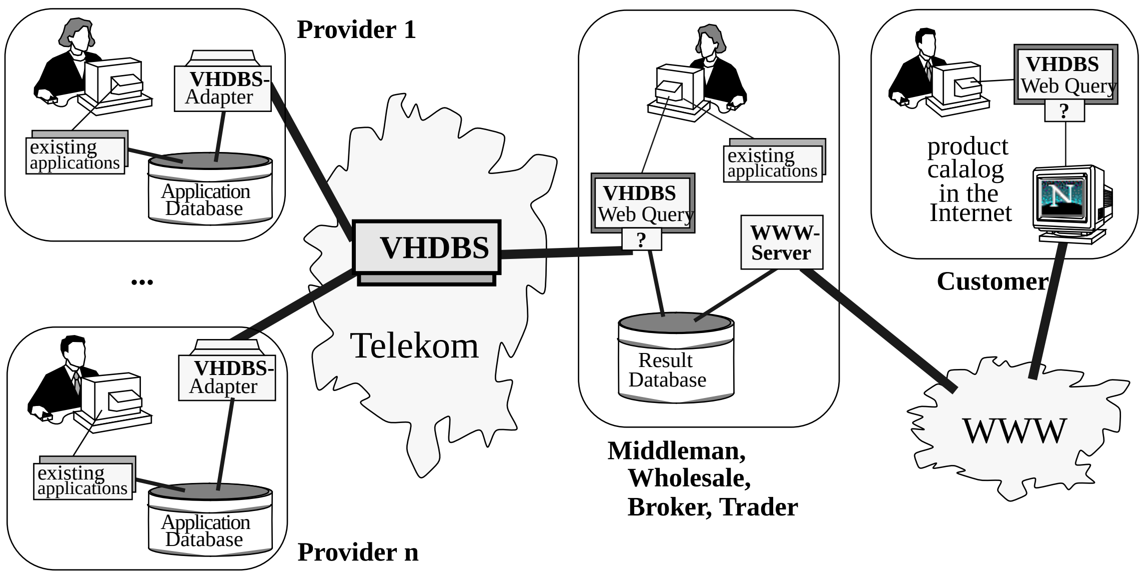

Figure 2: Using VHDBS for Integrating Business Partners

3. VHDBS Technical Overview

VHDBS enables the integration of heterogeneous data sources from existing database systems into a distributed enterprise-wide federated information system, and makes heterogeity and distribution transparent. This facilitates the reuse of existing functionality and information. Besides autonomous use of databases, a comfortable homogeneous object-oriented cooperated access to distributed heterogeneous legacy data and the combination of such data by inter-database queries is supported. VHDBS has the following properties:

- it supports the integration of external heterogeneous data sources or legacy systems
- data sources are accessed via adapters and remain autonomously accessable
- data migration is replaced by virtual integration, i.e. dynamic access to external data (data migration is also supported since VHDBS provides creation of new repositories)
- the federated system is object-oriented, but data sources of any kind can be integrated
- an object-oriented data model serves as mediator for the cooperation of all data sources
- data model, data definition language and query language are based on industry standards like OMG/CORBA and ODMG 2.0.

3.1. Architecture

Figure 3 illustrates the multi-layered architecture of VHDBS, consisting of several components, which can be distributed over several heterogeneous hosts and which cooperate via CORBA2 [OMG96a]. The underlying schema architecture corresponds to [SL90].

VHDBS provides three interactive interfaces: the FCLI command-line interface, the FGRAPH graphical user interface and the FWEB WWW interface. FCLI provides simple access to the system, and FGRAPH includes administrative functionality, while FWEB is for the regular user, and thus the main visible component in eCommerce applications. All interfaces access the VHDBS system through its API (application programming interface), which can be used directly by any application program.

The *federated server* is the kernel component. It aggregates all services provided at the federated layer (resulting in the *VHDBS-API*) and coordinates all servers of the federated layer, consisting of the schema subsystem (ODL parser and meta server), and the query subsystem (OQL parser, federated optimizer and query interpreter). The *meta server* manages meta data, which is used to provide the information about definitions of schemas/types, repositories, databases and users. Meta data is important for supporting cooperative use of distributed and heterogeneous component databases and is mainly entered via the *ODL parser.* Additional meta data can be entered via FGRAPH.

Federated queries are formulated in standard ODMG/OQL, which has been slightly extended for high-level descriptive manipulation, while distribution remains transparent. Queries are analysed by the *OQL parser*, and correct queries are sent to the federated optimizer and then to the query interpreter by the federated server. The *optimizer* mainly minimizes data transfers. Without optimization, the performance of federated queries is only acceptable for small amounts of data. Moreover, the optimizer treats several mappings including name mappings which are defined in OQL (this will be described in another paper). The *query interpreter* splits the optimized query into sub-queries for all related component databases, which are executed by them in parallel through the database adapters. All partial results are imported into the *Fed-DB*, and here they are combined to the final answer under control of the query interpreter (see [Wu97] for our early work on query processing).

At the bottom layer there are the component databases (*CDBs*), plugged into the system via database system adapters (*DBMS adapters*) situated at the next higher level. Besides the DMBS adapters for all component databases there is an adapter for the *Fed-DB* (storing intermediate results) and one for the *Meta-DB*, to be able to execute queries on meta data. Fed-DB, Meta-DB with their adapters and the federated server together form the VHDBS service, which may be provided by the Telekom.

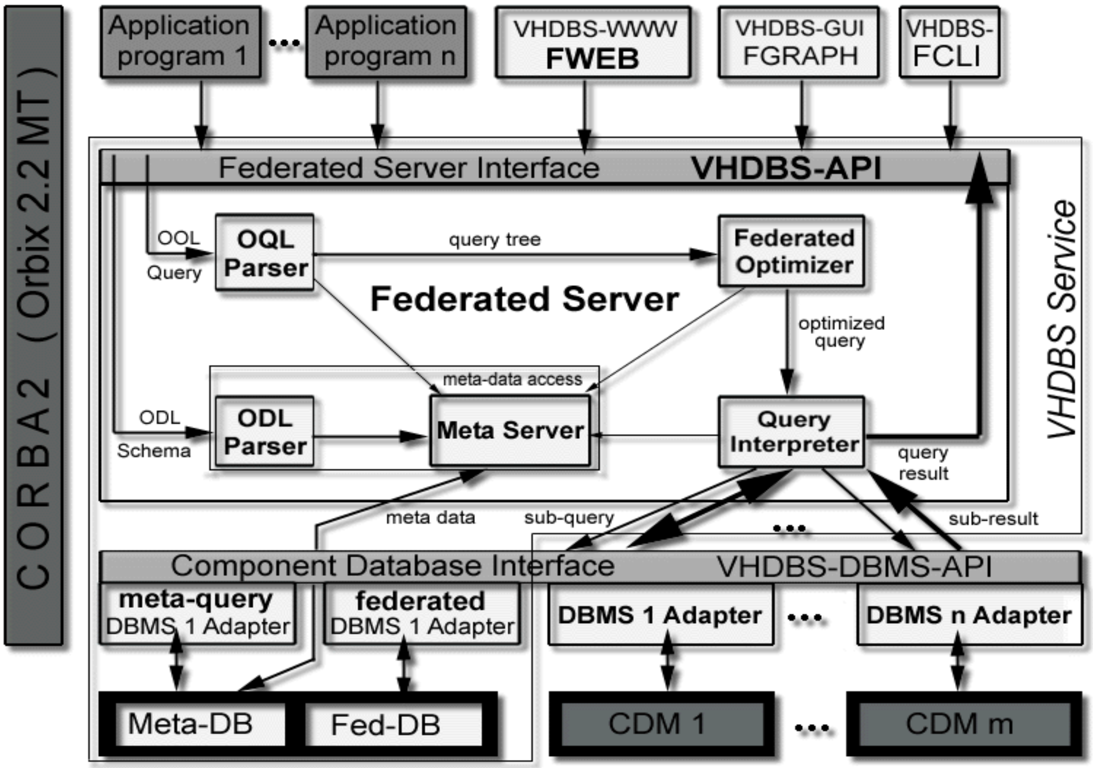

Figure 3: Architecture of the VHDBS System

A uniform interface *DBMS-API* is defined for all database adapters, regardless of their database architecture, and also used for Fed-DB and Meta-DB. This interface provides abstract operations to access the data within any component database. Each DBMS adapter for a specific database system (or system class) maps the federated data model to the data model of the component database system.

OMG/CORBA, an industrial standard for distributed object computing, is included in the architecture to support distribution and communication of all components. Our prototype is being developed using multi-threaded IONA/Orbix 2.2 [Iona97a]. All components of the architecture and all data objects described by the federated schema are specified using the IDL interface definition language of CORBA.

Each component may be instantiated multiple times to scale the VHDBS system. There may be clients connecting to their own federated server and others sharing a federated server. Each federated server instantiates all lower-level servers including the DBMS adapters. The open, modular, multi-tier architecture of VHDBS enables a transparent, flexible distribution of objects and data across a heterogeneous network platform. The system is flexible, scalable and extensable, both at database level and at schema level.

3.2. Data Model and Repositories

The data model serves as a basis for the interoperation concept of the system architecture. In VHDBS an object-oriented data model (*ODM*) is used, which is based on existing industry standards such as the object model of OMG [Bak97] [OMG96a] and ODMG2.0 [ODMG97]. This model provides a high level of abstraction and the most necessary concepts for modeling different data, so that it is easy to map data models of existing database systems and database applications to ODM and vice versa, thus making heterogeneity transparent for the user.

Objects in VHDBS are typed. There are two sorts of types: simple types and complex types. Simple types are types such as Integer and String, while complex types are based on different type constructors such as Tuple, Set and List. A more detailed description of the type concept is given in [Wu91] and [Wu96].

The concept of *repository* is key to VHDBS: Objects are both organized and stored in repositories. Queries are formulated based on repositories and query results are put into repositories. Thus, repositories are the virtual and logical storage units for objects at federated level and the entry points to access objects. By means of repositories, VHDBS provides materialized views on object sets originally stored in several component databases.

There are two kinds of repositories: mirror repositories and combi-repositories, i.e. combined repositories. By means of *mirror repositories,* the definitions (not the contents) of physical repositories of component databases are partially mirrored to the federated level. Physical repositories are storage units of component databases, which include relations in relational databases and entries or named objects within object-oriented databases.

Combi-repositories are defined by combining some existing repositories, which could be mirror repositories or themselves combi-repositories. Thus, a combi-repository can be used to define a rather complex relationship between the objects stored in some component databases. To specify such a relationship, restrictions and conditions can be given, using queries formulated in OQL.

To increase flexibility, there are different definition modi for repositories: firstly, the definition and/or the contents of a repository may be temporary or persistent. Secondly, repositories may be declared at two different schema levels (federated schema and external schemas in the 5-level-schema-architecture of [SL90]), and thirdly, a combi-repository may behave either as a “macro“ (its reference being textually replaced by its definition) or as materialized views (filled on direct or even indirect repository access).

Everything to be done with data objects is described by operations defined in the type interface. Operations are implemented as methods, which are either stored in component databases or implemented at federated level, either using Tcl [Wel94], OQL (via parameterized repository definitions), or (later) using any language for which a CORBA binding is available. Via method implementations arbitrary components can be integrated to be used by applications.

3.3. Web-based Query Interface

The FWEB Web interface of VHDBS allows the user to query and browse VHDBS repositories by means of a graphical user interface. The user may submit a query to select relevant subsets of the data and then browse these subsets by means of a graphical representation or use these subsets as a basis for a new query. Thus, the user can move back and forth seamlessly between queries and browsing activities. FWEB provides the following functionality:

- Login to VHDBS
- Commented listing of all information (mainly repositories) available to that user
- Formulating of federated queries (which equals defining combi repositories)
- Executing federated queries (filling of combi-repositories)
- Displaying query results
  - browsing and navigating in the structured object graph
  - executing of methods on selected objects, e.g. to display multimedia attributes or to manipulate objects.

The main functionality of FWEB is performing federated OQL queries and browsing their results. As only a few Internet users are familiar with OQL, FWEB supports pre-formulated parameterized queries, of which the user only sees a name and an informal description. By clicking the name, the query is executed. If it has parameters, the user is requested to provide actual values, and then the hidden OQL query is sent to the federated server, which calculates the results and sends them back to the user. These results may consist of highly structured information according to the ODL data model and application schema. These structures can be browsed and may contain embedded multimedia parts.

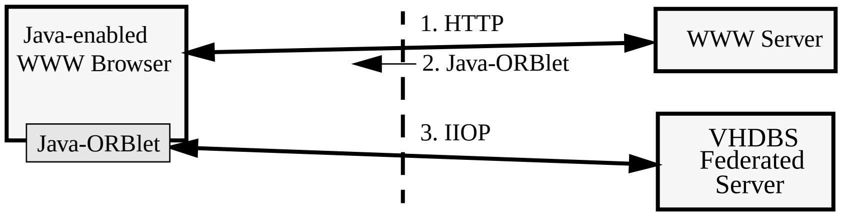

Figure 4: Communication Links of the VHDBS-Web Interface

FWEB is implemented in Java [Fla97][Eck97], and uses a Java-ORB (an ORBlet constructed using OrbixWeb 3.0 [Iona97b]) to connect to federated servers. As shown in Figure 4, when loading the FWEB HTML page into a Java-enabled Web browser, the Java ORBlet is loaded together with the first FWEB applet. After VHDBS login the next applets are loaded and the user gets a commented list of all federated repositories available to him.

We planned to provide the GUI FGRAPH [WW97] directly to Web users. FGRAPH is implemented in Tcl/Tk [Wel94] based on a C++ ORB. By using a Java-ORBlet and the Netscape Tcl-Plugin [Sun97] or a Tcl interpreter written in Java (see SunScripts Jacl and TclBlend [Sun97]) FGRAPH could be used via the Web, however it is only suitable for database experts and too complex for the regular Web user.

4. A Sample Application

For the application domain depicted in Figure 2, a VHDBS sample application in the plant market is outlined in Figure 5: A wholesaler or trader/broker composes his product catalogue via VHDBS access to all his "providers", condensing, combining and updating all information found. This is achieved by defining a federated repository using VHDBS and the query language OQL, which combines information exported by the providers. The broker copies the result into his local catalogue and provides it to potential customers using standard Internet techniques or again VHDBS.

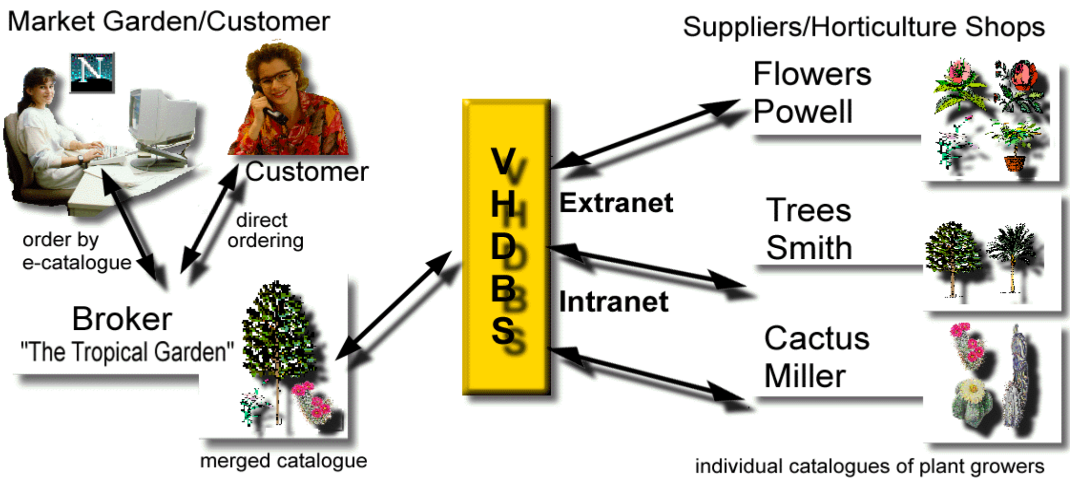

Figure 5: Integration of Business Partners in the Green Market

This scenario can be implemented using VHDBS as follows: (1) the federated schema including all interfaces and mirror repositories is defined, (2) the federated schema is homogenized with the help of combi-repositories, and (3) some queries (i.e. combi-repositories) can be provided to Web users via the external schema:

Firstly, each plant grower exports his product catalogue to VHDBS. For this purpose the federated schema including all mirror repositories is described in (slightly extended) ODL, a superset of OMG/IDL, reflecting the structure of, e.g., database tables for offered articles, used at that location or enterprise. Here, object-oriented features like inheritance can be used:

```
INTERFACE Article {
  ATTRIBUTE STRING  artNo;          // typical attributes in product
  ATTRIBUTE STRING  name;           // ... catalogues
  ATTRIBUTE LONG    price;
  ATTRIBUTE LONG    amount;
  ATTRIBUTE SET<STRING> properties;// in case of an OO database
  VOID      put_to_shopping_box(); // federated methods for broker
  VOID      order (IN STRING name, address, ...);
};
INTERFACE ArticleMM : Article {     // article with multimedia data
  ATTRIBUTE Multimedia  picture;
};
INTERFACE ArtMM {
  ... // not shown here, similar to ArticleMM, but other attribute names
};

REPOSITORY SET<Article>   Trees_Smith;     //
REPOSITORY SET<ArticleMM> Flowers_Powell;  // mirror repositories
REPOSITORY SET<ArtMM>     Cactus_Miller;   // of all suppliers
```

For each plant grower, such a type and mirror repository description as sketched above is provided. If some providers use homogeneous data schemas, the interfaces need only be provided once to VHDBS. In our example, *Powell* and *Miller* use similar schemas in their local databases, but some attribute names and the order of attributes differ, while *Smith* does not provide any pictures for his trees.

Secondly, combi-repositories are defined, completing the federated schema, and at this step heterogeneous data can be homogenized and horizontal and/or vertical fragmentation of overall logical data is expressed and made transparent. In our example, the broker, *Tropical Garden*, selects interesting offers from all his providers. Slightly simplified, this is:

```
INTERFACE MyArticle : ArticleMM {
  ATTRIBUTE LONG   orig_price;
  ATTRIBUTE STRING orig_number;
  MyArticle compose_offer (IN Article offer2, offer3); // see below
  BOOLEAN   isCompatible  (IN Article plant2, plant3); // see below
};

REPOSITORY MyTrees IS
  “SELECT MyArticle(T+artNo, name, 1.1*price, amount, properties,
                    no_picture, price, artNo)
   FROM   Trees_Smith WHERE <condition1>“;

REPOSITORY MyFlowers IS  // includes name mapping
  “SELECT MyArticle(T+no, flow_name, 1.05*Price, 1, description,
                    pict, Price, no)
   FROM   Flowers_Powell WHERE <condition2>“;

REPOSITORY MyCactus IS
  “SELECT MyArticle(T+artNo, name, 1.1*price, amount, properties,
                    picture, price, artNo)
   FROM   Cactus_Miller WHERE <condition3>“;

REPOSITORY MyPlants IS “MyFlowers UNION MyTrees UNION MyCactus“;
```

The first three definitions homogenize all data to type *MyArticle*. Here, only relevant articles are selected via any optional conditions, and new prices and article numbers are calculated by the broker. These repository definitions will typically be defined as “macros“, i.e. their definitions are inserted at the place where they are referenced. The last definition expresses the horizontal fragmentation (via unions) of overall logical data.

Thirdly, the broker may either migrate this data into an Internet storefront, or provide VHDBS Web access. For the latter purpose, parameterized queries over global data can be predefined or defined on-the-fly in the external schema, e.g.:

```
DEFINE Special_Offers (Property, PriceLimit) AS
SELECT Article(artNo, name, price, amount, properties, picture)
                    // orig_price and orig_number are hidden
FROM   MyPlants WHERE  Property IN properties AND price < PriceLimit;
```

Schema-specific search functions can be provided to Web users in this way by predefined queries, and the result is saved in the Fed-DB to be browsed using FWEB. For example, if a Web user clicks on *Special_Offers*, he is asked to enter properties and price limits of searched plants, and after specifying these attributes, blocks of plant records including pictures (if provided) are transferred to him and can be browsed. The display type *Article* used here may provide methods which the Web user can activate on selected objects, e.g. to submit orders or fill a shopping box. The broker *Tropical Garden* may also provide a second repository to his Web customes, which automatically combines some plants of different kinds, based on some common properties, as indicated in Figure 5, e.g.:

```
DEFINE Special_Combinations (season) AS
SELECT f.compose_offer(t,c)
FROM   f IN MyFlowers, t IN MyTrees, c IN MyCactus
WHERE  f.isCompatible(t,c) AND season IN f.properties;
```

Here, operations defined on data type *MyArticle* above are used to combine the plants.

5. Conclusion

In this paper we discussed the use of VHDBS for eCommerce applications. The federated database system VHDBS has been developed by an industrial research project, based on OMG and ODMG standards. An object-oriented data model is used as common data model (being adequate to integrate heterogeneous data) and a CORBA Object Request Broker is included in the system architecture. Several mapping layers of the architecture allow for an open system, which provides a high degree of scalability.

The Web user interface of VHDBS provides means for executing federated inter-database queries to combine heterogeneous data sources. Such queries may be formulated in ODMG/OQL, but normal Web users will prefer executing pre-fabricated parameterized queries, thus OQL is hidden to them. The queries are submitted via CORBA/IIOP to a federated server, which decomposes the query into partial queries for plugged-in component databases and then combines their partial results to form the final answer delivered to the user. The object browser of the VHDBS Web interface supports navigation within deeply structured object graphs and execution of methods on selected objects (e.g. to display multimedia attributes) or to use the result in subsequent queries.

The current version of the VHDBS system is a demonstrator of our idea of cooperative access to several distributed and heterogeneous databases, providing adapters for relational and object-oriented database systems, represented by Oracle [Ora92] and O2 [O2-95]. The WWW interface and other new parts in the architecture, like the optimizer, are under development. In the next phase of the project, the system will be enhanced towards transaction and security mechanisms, efficiency and global data consistency.

References

[Bak97] S. Baker. *CORBA Distributed Objects, Using Orbix*, Addison-Wesley, 7/1997

[BE96] O.A. Bukhres, A.K. Elmagarmid (eds). *Object Oriented Multidatabase Systems: A Solution for Advanced Applications*. Prentice Hall, Englewood Cliffs, N.J., 1996

[BFN94] R. Busse, P. Frankhauser, E.J. Neuhold. *Federated Schemata in ODMG*. In Proc.2nd Int.East/West Database Workshop, 1996

[BFH+94] R. Busse, P. Frankhauser, G. Huck, W. Klas. *IRO-DB, An Object-Oriented Approach Towards Federated and Interoberable DBMS*. In Proc. Int. Workshop on Advances in Databases and Information Systems (ADBIS'94), Moskow, Russia, 1994. See http://este.darmstadt.gmd.de:5000/dimsys/irodb/home.html

[Car95] M. Carey et al. *Towards Heterogeneous Multimedia Information Systems: The Garlic Approach*. In Proc. IEEE Workshop on Research Issues in Data Engineering, 3/1995

[Chu90] C.W. Chung. *Dataplex: An Access to Heterogeneous Distributed Databases.* Communications of the ACM, 33(1), 1990.

[Con97] S. Conrad. *Föderierte Datenbanksysteme - Konzepte der Datenintegration*, Springer Informatik, 1997

[Eck97] B. Eckel. *Thinking in Java*. MindView Inc., 11/1997, Prentice Hall (to appear). See http://www.EckelObjects.com/javabook.html

[Fla97] D. Flanagan. *Java in a Nutshell*, O'Reilly, 1997

[HST97] T. Härder, G. Sauter, J. Thomas. *Design and Architecture of the FDBS Prototype INFINITY.* In S. Conrad, W. Hasselbring, A. Heuer, G. Saake (eds): Engineering Federated Database Systems (EFDBS97), Proc.Int. CAiSE97 Workshop, Computer Science Preprint 6/1997

[HBP93] A.R. Hurson, M.W. Bright, S.H. Pakzad. *Multidatabase Systems - Advanced Solution for Global Information Sharing*. IEEE Computer Society Press, 12/1993

[IBM97] IBM. *Internet Business Opportunities with IBM Net.Commerce*. See http://www.internet.ibm.com/commercepoint/net.commerce/lit.html

[Int97] INTERSHOP Communications Inc. *White paper on Intershop Online*. See http://www.intershop.de/products/online/whitepaper.html, 5/1997

[Iona97a] IONA Technologies. *Orbix 2.2 Programming Guide*. 3/1997

[Iona97b] IONA Technologies. *OrbixWeb 2.2 Programming Guide*. 11/1997

[KFA94] W. Klas, G. Fischer, K. Aberer. *Integrating Relational and Object-Oriented Database System using a Metaclass Concept.* Journal of Systems Integration, Vol. 4 No. 4, 1994

[Kim93] W. Kim et al. *On Resolving Schematic Heterogeneity in Multidatabase Systems*. Journal of Distributed and Parallel Databases, Vol. 1, 1993

[Kim95] W. Kim (ed). *Modern Database Systems: Object Model, Interoperability, and Beyond*. ACM Press/Addison Wesley, 1995

[LMR90] W. Litwin, L. Mark, N. Roussopoulos. *Interoperability of Multiple Autonomous Databases*. ACM Computing Surveys, Vol. 22, 1990

[Mid96] Open Database Middleware Project Team. *OpenDM System Overview.* C-LAB Cooperative Computing & Communication Laboratory, 1996. See also http://www.c-lab.de/~opendm/help.html.

[ODMG97] R.G.G. Cattell el al (ed.). *The Object Database Standard ODMG 2.0*. Morgan Kaufmann Publishers, 1997

[OMG96a] OMG. *The Common Object Request Broker: Architecture and Specification*, Revision 2.0, Updated 7/1996

[OMG96b] OMG. *CORBAservices: Common Object Services Specification*, Updated 11/1996

[Ora92] Oracle Corporation. *Oracle 7 Server, SQL Language Reference Manual*. Part No.778-70-1292, 12/1992. See also http://www.oracle.com/

[ÖV91] M.T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 1991

[O2-95] O2 Technology. *O2Tools User Manual*. O2 Technology, 1995. See http://www.o2tech.com/

[SAD+95] M.-C. Shan, R. Ahmed, J. Davis, W. Du, W. Kent. Pegasus: *A Heterogeneous Information Management System*. In [Kim95]

[SAL+96] M. Stonebaker, P.M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, A. Yu. *Mariposa: a Wide-aera Distributed Database System*. VLDB Journal, 5(1), 1996

[SL90] A. Sheth, J. Larson. *Federated Database Systems for Managing Distributed, Heterogenous and Autonomous Databases*. ACM Computing Surveys, 22(3), 9/1990

[Su96] S.Y.W. Su et al. *NCL: A Common Language for Achieving Rule Based Interoperability among Heterogeneous Systems*. Journal of Intelligent Information SystemsVol. 6, 1996

[Sun97] Sun Microsystems Laboratories, SunScript group. *Introducing Jacl and Tcl Blend*. See http://sunscript.sun.com/, 1997

[TTC+90] G. Thomas, G.R. Thomson, C.-W. Chung, E. Barkmeyer, F. Carter, M. Templeton, S. Fox, B. Hartman. *Heterogeneous Distributed Database Systems for Production Use*. ACM Computing Surveys, 22(3), 1990

[UA95] S. Urban, T.B. Abdellatif. *Object-oriented Query Language Access to Relational Databases: A Semantic Framework for Query Translation*. Journal of Systems Integration, Vol. 5 No. 5, 1995

[Wel94] B. Welch. *Practical Programming in Tcl and Tk*. Prentice Hall, 1994.

[Wu91] X. Wu. *A Type System for an Object-Oriented Database System*. In Proc. 15th Annual International Computer Software & Applications Conference, Tokio, Japan, September 11-13 1991. IEEE Computer Society Press.

[Wu96] X. Wu. *An Architectural Framework for Interoperation of Distributed Heterogeneous Database Systems*. In Proc.2nd IEEE Int. Conf. on Engineering of Complex Computer Systems (ICECCS), Montreal, IEEE Computer Society Press, 1996

[Wu97] X. Wu. *An Approach to Query Translation in a Federation of Distributed Heterogeneous Database Systems*. In Proc. 4th Int. Conf. on Object-Oriented Information Systems (OOIS), Brisbane/Australien, Springer-Verlag, LNCS, S.55-65, 11/1997

[WW97] X. Wu and N. Weissenberg. *A Graphical Interface for Cooperative Access to Distributed and Heterogeneous Database Systems*. In Proc. Int. Conf. IDEAS, IEEE Computer Society Press, 1997.

[ZCF+97] C. Zaniolo, S. Ceri, C. Faloutsos, R. Snodgrass, V.S. Subrahmanian, R. Zicari. *Advanced Database Systems*. Morgan Kaufmann Series in Data Management Systems, 5/1997