Interoperability Issues in Time Series Management

Dreyer Werner
UBILAB, Union Bank of Switzerland
dreyer@ubilab.ubs.ch

Abstract

Time series management is a very important task for financial institutions like banks. It serves as a basis for services like investment research or portfolio management. A time series management system (TSMS) has to provide specific data management functionality [DKS93]. An important aspect of a TSMS is interoperability between different time series bases and between time series bases and external data sources or client applications. These components are highly heterogeneous and autonomous, so that integration is difficult to realize. In this paper, we describe which problems this integration poses, and the direction our solution is heading into. The data exchange is influenced by several relevant factors: The source object, the target object, the data model, the data schema, the data location, the consistency requirements, the scheduling of the data exchange, the exchange facility, and the communication mechanism. The goal of our project is to set up data exchange automatically, based upon a declarative specification mechanism.

1 Introduction

The paper is organized as follows: Chapter two gives an overview over related research work. Chapter three describes the TSMS project in general and its interoperability aspects in particular. Chapter four introduces a model of data exchange within the TSMS and between the TSMS and external sources or client applications. Chapter five explains the concepts of our solution. Chapter six makes a conclusion.

2 Related Research

2.1 Heterogeneous Database Systems

We distinguish between tightly and loosely coupled systems [SL90]. There are at least three topics in the area of tightly coupled systems that are also of interest to our project: Global data models [BND90, PM88], schema analysis, schema integration [BLN86, SH92, SM91]. Loosely coupled systems have more in common with our project than tightly coupled systems. One topic to study are multidatabase languages [LMR90]. They may help to specify the data exchange between two heterogeneous components. Loosely coupled systems also deal with the maintenance of data consistency between autonomous components.

2.2 Work Flow Management Systems

A work flow is an activity that involves multiple related tasks executed by different entities [Shet93]. Work flow activities usually have to deal with different databases on different systems, so they face problems similar to ours.

2.3 Distributed Systems

There is ongoing research in the domain of toolkits for distributed systems, e.g., Arjuna or ELECTRA [Maff93]. They are supposed to ease the development of distributed systems, e.g., by providing fault tolerance or location transparency.

2.4 Data Formats

A possible topic of interest are data formats which are specifically suited for data exchange. There exist many proposals for such formats. They are mainly designed for specific application classes, e.g., CAD. We will check whether there are formats that are suitable for our application domain. In the EDI (electronic data interchange) area, for example, there are currently many ongoing efforts for the standardization of data interchange, like

This paper was published in the Proceedings of the Workshop on Interoperability of Database Systems and Database Applications; Fribourg, Switzerland, Oct. 1993, pp. 235-246.

EDIFACT for commerce, or SWIFT for data interchange between banks.

A second issue in this area are self describing data formats like "Standard Format Data Units" (SFDU) [LMR90] or "Flexible Image Transport System" (FITS) [Well]. Self describing data formats require no global schema and preserve component autonomy. SFDU incorporate a reference to a meta object that is known in the entire system and that can interpret the data of the unit.

3 The Time Series Management System Project

3.1 Time Series Management as a Special Purpose Application Domain

Time series are a very important kind of data for economic, financial and other scientific research activities. At Union Bank of Switzerland (UBS), several departments work intensively with time series, e.g., the Economic Research Group, Asset Allocation and International Portfolio Management, or Broker Research. Within the whole bank, there are several thousands of time series used. This makes their management a difficult task. Different departments may collect the same time series. Moreover, it is problematic to find the time series relevant to a certain task. Up to now, time series have been stored in files or in general purpose database systems. This solution turned out to be inappropriate. Therefore, UBILAB, the information technology laboratory of Union Bank of Switzerland, decided to develop a TSMS, i.e., a database system that is especially tailored to handle time series data. It will have the following characteristics:

- Its data model, i.e., the structural elements and the functional capabilities, is exactly designed for time series data.
- It provides functionality for statistical data quality management.
- It allows to import data from various kinds of external data sources, exchange data between different time series bases, and export data to various client applications.

In the rest of this paper, we will only treat the last aspect, i.e., data exchange. A more detailed description of the TSMS project in general can be found in [DKS93].

A time series management system has to deal with many different, heterogeneous components. The autonomy of these components has to be preserved, because they are supposed to be used as independently as before. These reasons led us to the conclusion that we have to build a loosely coupled system. The fact that we only have to deal with a very specific application domain and therefore with rather uniform data reduces the complexity of our system.

3.2 Overview of the General System Architecture

Figure 1 shows the general system architecture of the time series management system and its environment.

Our TSMS is a system of time series bases that exchange data between each others. Some of these time series bases also import data from external sources, and others exchange data with client applications.

3.3 Components

3.3.1 External Data Sources

The most important external data sources are commercial data providers and central databases within the bank. Data from commercial data providers can be received as files, via online data streams, or they can be imported from online databases. An example for file delivery is HIKU (historical courses) from the Swiss company Telekurs. Selectfeed from Reuters is an example for an online data stream. Online databases are, e.g., operated by the WEFA group. They are autonomous and entirely read-only. The connection is often based on packet switching services. Central databases also play an important role in the TSMS environment. At UBS, a central database stores a huge quantity of financial data.

These external data sources have the following characteristics:

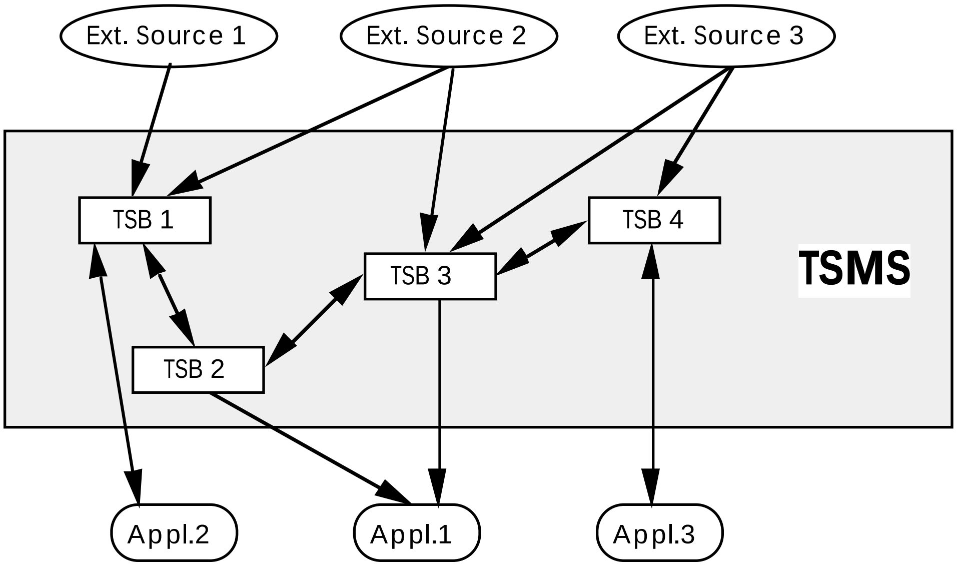

Fig. 1: General System Architecture

- Their data model, data schema, and data format is usually different from the TSMS.
- We cannot modify the interface of these components. E.g., there is normally no way to get notified about changes in the data.
- Their access and query possibilities differ very much.

3.3.2 Time Series Bases

The TSMS contains many different time series bases. There are public time series bases, work group bases, and individual bases. An example for a public time series base is an OECD database that stores all the OECD data imported from an external source. A public time series base is accessible for all researchers. A work group time series base belongs to a whole team of researchers. Most of the data processing is done in individual time series bases; work group and public databases serve mainly as a source of raw data. Researchers get data from a work group or public data base, process them in their individual base, and store them again in a work group or public base, where they serve as new raw data for some other work.

The different time series bases use the same data model (the TSMS data model) and data format, but they normally have different schemata, as defined by the individual researcher.

3.3.3 Applications

Time series are processed by many different tools, e.g., statistic packages, spreadsheets, and desktop publishing programs. These tools all use their individual data model, format, and schema.

3.4 Data exchange

There are three categories of data exchange in our system:

- Data import from external sources into a time series base.
- Data exchange between different time series bases.
- Data exchange between time series bases and applications.

3.4.1 Data Import from External Sources

There are no powerful facilities that guarantee consistent data exchange (e.g., no wrong or double data) between these external sources and the TSMS, we can just import data from these sources. This makes data consistency requirements more difficult to fulfill. Take, for example, data import from an online data stream: We must be able to import consistent time series, even if the connection breaks down after a partial transmission, and the transmission is restarted somewhere before the last transmitted data.

A further problem is the variety of communication means: There are low-level facilities like DCE-RPC, Internet services like FTP, distributed system approaches like CORBA, electronic mail, X.25, simple dial up phone lines, or even data exchange via disk/tape. The various communication means heavily influence the possible transaction concepts.

Data import from external sources has to take into consideration that the data model, the schema and the format of these sources are different from those of the TSMS.

Normally, users do not import data directly from external sources. In general, there is, e.g., a time series base for OECD data that imports the data from the external sources. A researcher who needs OECD data gets them out of the time series base where they are already in the appropriate data model and data format.

3.4.2 Data Exchange between Different Time Series Bases

Data exchange between time series bases is bi-directional, in contrast to the previously explained data import from external sources. Researchers may get raw data out of the time series bases of their colleagues, work with these data, e.g., by adding new estimations, and then restore them. This means that we will have to provide more powerful transaction facilities than for unidirectional data import.

We will have to define the consistency requirements for the data exchange that we can fulfill between different time series bases. This also depends on the available communication means. Because the different time series bases often are in the same LAN, communication means between them will often be more powerful than between time series bases and external sources, but there may also be low level connections like dialup phone lines.

Another point of interest is the scheduling of the data exchange: Data exchange is either unique or subscribed.
Data exchange between time series bases is simpler than import from external sources, because they do not differ in the data model and the data format, but only in the data schema.

3.4.3 Data Exchange between Time Series Bases and Client Applications

This kind of data exchange may be unidirectional, e.g., from a time series base to a desktop publishing or charting program, or bi-directional, e.g., a statistics package gets data from a time series base, processes them, and feeds them back into the time series base.

Consistency requirements may vary heavily. When we export data to a charting tool, we only have to deal with correct data exchange. When we work with a statistics package that imports data, processes them, and transfers them back into the time series base, we also have to maintain semantic correctness of the data.

Like external data sources, client applications all have their own data model, schema, and format.

If some applications provide the necessary interfaces, an import and export facility via mechanisms like DDE would be an interesting issue.

3.5 Goals regarding Interoperability

Our TSMS project has the following goals regarding interoperability:

- Researchers must be able to work with data from different sources, whether they are databases, files, or data streams.
- The TSMS must allow the export of data to client applications like charting or statistics packages, and data processed in such an application can be stored back into a time series base.
- The different sources use different schemata or even different data models. Once the connection between two sources is set up, the users should no longer have to distinguish where the data are located, which underlying data model

they have and in what schema they are.

- The system should handle data exchange using various exchange facilities and communication mechanisms.
- Users must be able to schedule the data exchange according to several criteria.
- The system must be able to deal with components that are loosely coupled and highly autonomous.

4 A Data Exchange Model

4.1 Overview of the Data Exchange Model

The data exchange model consists of four components:

- *The source object*: It describes the data to be transferred and its relevant environment from the point of view of the data producer.
- *The target object*: It describes the data to be transferred and its relevant environment from the point of view of the data consumer.
- *The exchange facility*: The exchange facility is the way two components exchange data. This can be direct program to program communication, indirect communication via file transfer, or a distributed database system.
- *The communication mechanism*: This component describes the technical means that are available to realize the exchange facility, like electronic mail or X.25. They can be from different abstraction levels: We can use RPCs directly, or we use electronic mail, which is in turn implemented with RPCs.

All these components influence the way data are exchanged by different factors. Figure 2 gives an overview over this situation.

4.2 Factors determining the way data is exchanged

4.2.1 Source Object, Target Object

The specification of the source and target objects indicates the data to be exchanged. This description varies heavily, depending on the kind of system the data object is stored in. An example is the import of a time series from a relational database system into a time series base. The source object is described with an SQL-query, e.g., "select price from timeSeriesTable where name = "UBS" and year > 1980", while the target object is expressed according to the TSMS-DML.

4.2.2 Data Model, Data Schema, Data Format

Data model, data schema, and data format are related terms. Data model means the logical constructs that are offered by a DBMS or an application to describe the data. An example are relations in the relational model. The data schema describes the way we structure the data according to a given data model. There are, e.g., different ways to partition the information into relations of the relational models. The data format describes the representation of the data. This can be on a low level, e.g., the Byte ordering of an integer on different machines. Data formats can also concern a higher level, e.g., the ordering of the elements of a time series. The limit between data format on a high level and the data schema is fluid.

The different components (databases, external sources, or applications) are either homogeneous or heterogeneous, with different degrees of heterogeneity. Two components are homogeneous if they use the same data model, data schema, and data format, otherwise they are heterogeneous. When we exchange data objects between two homogeneous components, no transformation of these data is necessary; for data exchange between two heterogeneous components, however, we must specify a mapping that allows the data object transformation.

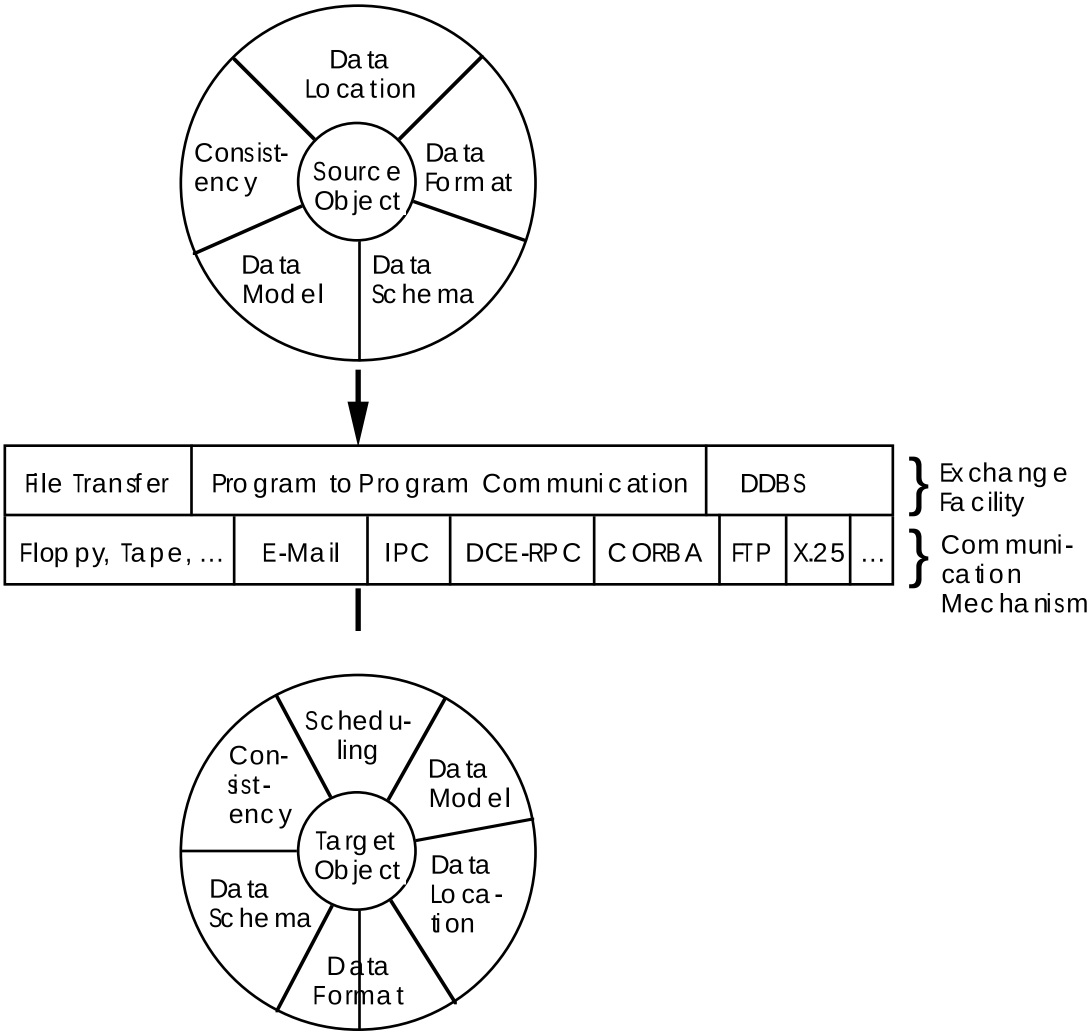

Fig. 2: Data Exchange Model

For this purpose, we will use techniques of schema analysis and schema integration, but we hope to be able to use rather simple methods, because we have very special purpose data.

To limit the number of necessary mappings, we will define a common data exchange format, so that we only need transformations between a specific and the common exchange format, but no transformations between specific formats. For the purposes of the TSMS, we develop a logical data model that is especially designed to handle time series data. We will have to find out whether this model is also suited for data exchange or not.

4.2.3 Data Location

The data location may be very different, depending on the kind of data source or target: When we exchange time series between two local time series bases, the location consists of the time series base. The import of a remote file may require an Internet address and a pathname of a file.

4.2.4 Consistency

Data is consistent when it is semantically correct. This is a very comprehensive goal. To reach it, the necessary considerations already have to be made during the logical design of the databases. As far as interoperability is concerned, we have to fulfill the following requirements:

- The system must be able to exchange data correctly, i.e., it has to recover from communication failures.
- Data has to be delivered in time: For example, a time series is imported into another time series base. Then, some new data values are appended to the master time series. Now we have inconsistencies between the master time series and its copy in the other time series base. We cannot avoid such situations in any case, and often a certain delay is tolerable. The degree and the duration of these inconsistencies depend on the applications. This fact is comparable with the problem of long transactions in engineering databases. There are three reasons why some changes are not transmitted fast enough: The data source cannot send the changes rapidly enough, the communication mechanism does not have enough band width, or the data sink is not able to absorb the transmitted data quickly enough. However, we believe that the absorption capability of the sink is rather high, because there is usually just one process writing a time series, and, furthermore, new data may be appended to a time series, but old data are seldom changed.

In a heterogeneous system with highly autonomous components, it is very difficult to maintain data consistency. The higher the autonomy of the components is, the more difficult it becomes to adhere to consistency requirements.

4.2.5 Scheduling

Users may have different requirements regarding the scheduling of a data exchange: It may be unique or subscribed.

Unique data exchange means that a user gets some data once and has no need for further updates, respectively he or she wants to specify explicitly when an import or update of some data takes place. This solution makes sense in two situations:

- A researcher does a data analysis just once. Take, for example, a portfolio manager who examines the stock prices of a specific company and then decides not to include these stocks in his portfolios.
- Data change very often, or new data sets are added all the time, but the user does not depend on the newest data. The data exchange costs are reduced in this way. This may be the case when the portfolio manager decides to examine the previously mentioned stocks again from time to time, but not at regular intervals.

When a user subscribes to some data, he or she receives the data for a first time, and also all subsequent changes or additions to those data. There are two variants of subscription:

- A researcher wants some data at regular intervals. This is useful when the data are delivered as files from data providers like Telekurs. Such companies offer their new data with regular frequency (mostly daily), and the time when the new data are available is known in advance. Another example is the estimation of the future inflation rate: Some base values like the increase of the money supply are published every month.
- Data exchange takes place when some specific event has happened. An example of such an event is the modification of a data value. This solution is chosen when the user always needs up-to-date data. To realize such a solution, the system providing the data must be able to notify its clients, and the client must be designed to handle such messages. An example is the notification of a researcher when a stock price drops below a certain threshold.

When importing data from another time series base or from an external source, the exchange may be unique or subscribed. For exporting data to a client application, only unique data exchange, i.e., exchange on demand, will be provided.

4.2.6 Exchange Facility

While we intend to realize program to program communication between different time series bases, there will be either file transfer or program to program communication between time series bases and the different external sources or client applications.

File transfer between external sources or client applications and time series bases is considered as a minimal solution that is always possible to realize when there are no communication facilities on a higher level. With many databases and applications, we can also have program to program communication, e.g., via mechanisms like Apple's Data Access Language (DAL), Microsoft´s Open Database Connectivity (ODBC) [Udel93b], or Dynamic Data Exchange (DDE). With DAL or ODBC, an application can use a database via SQL-statements. There is also a standard in this area, the SQL Access Group Call Level Interface (SAG-CLI). Data import from online data streams can also be considered as program to program communication, but on a very low level. There is only a very simple protocol that does not handle things like connection interruptions, etc.

Between different time series bases, we intend to realize program to program communication, or, depending on the DBMS we use to implement the system, we can use the facilities of a distributed database system.

Data exchange between time series bases and client applications is similar to data exchange between time series bases and external sources. File transfer is a minimal solution, but program to program communication via concepts like DDE is preferable.

Basically, the available exchange facility does not influence the possible degree of consistency that we can reach, but it is easier to realize with a distributed database system or at least program to program communication than with simple file transfer. A transaction concept that is implemented with file transfer may result in unacceptable performance degradation, so that we prefer to lower the consistency requirements.

4.2.7 Communication Mechanism

There are a variety of possible communication mechanisms, ranging from simple concepts like tape (or floppy disk, etc.) exchange to sophisticated architectures like CORBA.

Under certain circumstances, tape/floppy exchange may be the only communication mechanism between a time series base and an external source or client application.

Electronic mail is a rather limited, but highly available means of data exchange. We intend to provide a mail-oriented interface to our system, so that we can send and receive time series data via mail. For easier integration of mail services into applications, there will probably be more and more mail APIs. An example for such an API is MAPI (Microsoft´s Messaging API) [Udel93a] for Windows applications.

Interprocess communication [Stev92] is a technique for processes to exchange information. Under UNIX, there are several IPC-facilities, but most of them are only designed for communication between processes on the same host. Moreover, not all UNIX versions support the same IPC-facilities.

In a networking environment, there is the possibility of accessing data via client-server applications. The component that asks for the data is considered to be the client, and the data provider is the server. A well-known method to implement client-server applications is to use Distributed Computing Environment Remote Procedure Calls (DCE RPC) from Open Software Foundation (OSF) [RKF92, Shir92]. We plan to provide this access facility in our system.

The Common Object Request Broker Architecture (CORBA) from Object Management Group (OMG) [OMG91] has goals which are similar to DCE RPC, but on a higher level. DCE RPC implement remote function execution, while CORBA deals with the notion of objects. It allows to transparently send requests to remote objects and to receive their responses. Up to now, there are only very few CORBA implementations. If this changes during the development time of our project, we would consider to integrate data access via an Object Request Broker. Toolkits like Arjuna or ELECTRA are similar to CORBA. Most of these toolkits are ongoing research projects, and we will have to study the results to decide whether we can use them for our purposes.

Communication via file transfer is a possible solution when there is no online connection between the different systems. Two well-known standards for file transfer are FTP (File Transfer Protocol) and FTAM (File Transfer, Access and Management). FTP is a protocol which is implemented on TCP/IP, FTAM is a protocol conforming to OSI layer 7. We intend to incorporate data exchange via FTP in our system.

The available communication mechanism influences the choice of the exchange facility, too. Program to program communication is at least difficult to realize with communication mechanism like electronic mail or FTP, and it becomes very slow.

5 The targeted solution

We want to build a system that fulfills the goals mentioned in chapter 3.2. This system should ease the task of building up a connection between heterogeneous, autonomous components. Although our project is situated in the area of time series management, we hope to find a solution that is also applicable for similar kinds of problem domains. In order to be useful for several applications, we have to do the following:

- We will develop a specification formalism that describes all the parameters necessary for interoperability.
- Our solution will be incorporated in an application framework.

A specification formalism has to incorporate the following parameters, which are described in chapter 4:

- The source and the target object.
- The data model, the data schema, and the data format.
- The location of the data.
- Consistency requirements.
- The scheduling of the data exchange.
- The exchange facility.
- The communication mechanism.

6 Conclusions

Data exchange in a TSMS is a central aspect of the entire system, and it is emphasized much more than in other, common purpose database systems. The system described in this paper mainly exchanges data by means of replication. This is a rather pragmatic approach. The use of this system will show us whether this is sufficient, or whether a future version must be headed towards distributed transactions or a proxy mechanisms, so that we need as little data replication as possible.

References

[BLN86] Batini, C., Lenzerini, M., Navathe, S.: A Comparative Analysis of Methodologies for Database Schema Integration; ACM Computing Surveys, Vol. 18, No. 4, Dec. 1986.

[BND90] Böhnlein, P. G., Nittel, S., Dittrich, K. R.: Semantic Data Models; HMD Theorie und Praxis der Wirtschaftsinformatik, No. 152, Forkel-Verlag, Mar. 1990 (in German).

[Ceri84] Ceri, S.: Distributed Databases; McGraw-Hill computer science series, 1984.

[DKS93] Dreyer, W, Kotz Dittrich, A., Schmidt, D.: Research Perspectives for Time Series Management Systems; UBILAB Report, to appear in summer 1993.

[KS92] Karabatis, G., Sheth, A.: Specifying Interdependent Data: A Case Study at Bellcore; Bellcore Technical Memorandum TM-STS-021301/1, Jul. 1992.

[LMR90] Litwin, W.,Mark, L., Roussopoulos, N.: Interoperability of Multiple Autonomous Databases; ACM Computing Surveys, Vol. 22, No. 3, Sep. 1990.

[Maff93] Maffeis, S.: Object Oriented Distributed Programming with ELECTRA; Technical Report ifi-tr-92.23, University of Zurich, Institute for Computer Science, Jan. 1993.

[OMG91] Object Management Group: The Common Object Request Broker: Architecture and Specification; Draft 10, Dec. 1991.

[PM88] Peckham, J., Maryanski, F.: Semantic Data Models; ACM Computing Surveys, Vol. 20, No. 3, Sep. 1988.

[RKF92] Rosenberry, W., Kenney, D., Fisher, G.: Understanding DCE; O'Reilly & Associates, Inc., Sep. 1992.

[SH92] Sheth, A. P., Howard, M.: Schema Analysis and Integration: Methodology, Techniques, and Prototype Toolkit; Bellcore Technical Memorandum TM-STS-019981/1, Mar. 1992

[Shet93] Sheth, A. P.: Documentation of the course "Interoperability in Multidatabase Systems"; Swiss Federal Institute of Technology, Apr. 1993.

[Shir92] Shirley, J.: Guide to Writing DCE Applications; O'Reilly & Associates, Inc., Jun. 1992.

[SL90] Sheth, A. P., Larson, J. A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases; ACM Computing Surveys, Vol. 22, No. 3, Sep. 1990.

[SM91] Siegel, M., Madnick, S.: A Metadata Approach to Resolving Semantic Conflicts; Proceedings of the 17th International Conference on Very Large Databases, Barcelona, Sep. 1991.

[Stev92] Stevens, W.R.: Advanced Programming in the Unix® Environment; Addison-Wesley Publishing Company, Inc., 1992.

[Udel93a] Udell, J.: Simple MAPI Delivers; Byte, Apr. 1993.

[Udel93b] Udell, J.: Connecting Windows to Data with ODBC; Byte, Jan. 1993.

[Well] Wells, D.: FITS - A Self-Describing Table Interchange Format.

[WGM89] Weinand, A., Gamma, E., Marty, R.: Design and Implementation of ET++, a Seamless Object-Oriented Application Framework; Structured Programming 2:1-25, Springer Verlag, 1989.

[Zehn87] Zehnder, C. A.: Information Systems and Databases; Verlag der Fachvereine an den schweizerischen Hochschulen und Techniken, Zurich 1987 (in German).