Research Perspectives for Time Series Management Systems

Werner Dreyer, Angelika Kotz Dittrich, Duri Schmidt
UBILAB, Union Bank of Switzerland, Zurich
{dreyer, dittrich, schmidt}@ubilab.ubs.ch

Abstract

Empirical research based on time series is a data intensive activity that needs a data base management system (DBMS). We investigate the special properties a time series management system (TSMS) should have. We then show that currently available solutions and related research directions are not well suited to handle the existing problems. Therefore, we propose the development of a special purpose TSMS, which will offer particular modeling, retrieval, and computation capabilities. It will be suitable for end users, offer direct manipulation interfaces, and allow data exchange with a variety of data sources, including other databases and application packages. We intend to build such a system on top of an off-the-shelf object-oriented DBMS.

1 Introduction

At Union Bank of Switzerland, several departments work intensively with time series. They encounter a number of problems:

- The large volume of time series (several thousand) makes their management a difficult task.
- In large time series bases, researchers have problems finding the time series relevant to their work.
- When a time series base becomes too large, data quality management is impossible without the help of a DBMS.
- Researchers usually do not work with all the time series an institution collects. Instead, there is a need to build and maintain project-specific time series bases.
- Such a time series base must contain data from public and company databases as well as project-produced data.
- The same time series are used with many different tools, e.g. statistics packages, spreadsheets, and desktop publishing programs. Without the help of a DBMS that can cope with the different data formats these tools use, researchers are forced to duplicate the data.
- Researchers need specific system support for tasks like transforming the periodicity of time series, filtering, or computing new time series.
- As current TSMS are difficult to use, economists cannot work without the help of a database specialist.
- Synchronization problems may arise when time series are used concurrently by different users.

Research in time series management has to address these problems and to provide adequate solutions.

The remainder of the paper is organized as follows: First of all, we address the properties of TSMS. Then, we show an example of a time series base. We look at current solutions and at research related to our problem domain. Finally, we discuss the goals and the research issues of our project.

2 Requirements for TSMS

2.1 Data Model Requirements

2.1.1 Structural Elements

A data model for time series contains the following structural elements: Events, time series, groups of time series, other data, and time series bases.

Events

Events are the basic building blocks of time series. An event consists of the event data, which is time-variant. Examples of event data are the opening, low, high, and closing prices of a share. The event data can have an atomic or a structured data type. Atomic data types are scalar types like integer or string. Records or arrays of atomic types are examples for structured types.

Time Series

A time series consists of a header and a sequence of events ordered chronologically. Header data are shared by all events. They may be time-invariant and describe common properties of the time series as a whole (e.g. the location of the stock exchange), or they may be time-variant and derived from the events (e.g. the average closing price).

Normally, all the events of a time series are of the same type, but they can also vary over time.

Data values are either base or derived values. Base values record measured facts. Derived values are computed directly or indirectly from base values.

Data values differ in what they measure. One example are stock values, i.e. values which measure a value at some point in time, e.g. a price of a share. Stock values can be differentiated further. They may measure the value at the beginning or end of a period, or the highest, lowest or average value of a period. Another example are flow values. They measure a value over a period of time, like a cash flow. These different kinds of values have different periodicity transformations. An example is the transformation from a monthly to a quarterly periodicity: For the high selling price, one has to choose the maximum of the three monthly values, for the closing price the last value and for the cash flow the sum of the three values.

A calendar is associated with each time series. The calendar does the mapping between events and the time when the event occurred. In the case of time series with constant time between events, the calendar also defines the periodicity of the time series.

Time series differ in their density, i.e. in how many events are recorded. Missing events may arise from diverse reasons, such as that the event did not happen or the values of an event are not known.

A further property is the ordering of the events over time. The events of a time series may be completely ordered, i.e. there is at most one event per point in time. In contrast, the events of other time series may be only partially ordered, i.e. more than one event with the same time stamp may exist in the time series. An example for such a situation is the use of estimation methods for future values. A researcher may record several events per point in time, because he or she makes repeated estimations for certain data values after having received more accurate base data.

The cardinality of a time series, i.e. the number of events of a time series, varies with the application area. In the case of economic time series, the cardinality ranges from several tens in the case of annual data to several thousands in the case of daily data.

On the one hand, time series data are accessed along time, i.e. event sequences of one time series are examined. On the other hand, researchers also use cross-sectional analysis, whereby several time series are explored at a certain point in time. The data model must allow efficient storage and access methods for both ways.

Groups

Detecting time series in large time series bases relevant to the interests of a user is an important issue. One way to facilitate this is to partition the universe of all time series of a time series base into several groups according to various criteria, e.g. branches, country, or size of the company. A TSMS must support a flexible, powerful grouping mechanism.

A group consists of its header and its member set. The header contains data shared by all members, e.g. the group name, or data derived from some members, such as the covariance matrix of some time series contained in that group. The member set contains the time series and subgroups belonging to the group. It is necessary that a group can also contain other groups, that it is possible to define the members of a group by enumeration or by computation, and that members can belong to more than one group.

Other Data

Applications which use time series may also have to manage other persistent data that are not related to time series or groups. Examples are results of simulations or constants. These data are often not time series themselves. There are also meta data, i.e. data describing the actual events, time series, and groups. Unfortunately, there is no clear definition of what meta data are. Often, one user's meta data are the other user's data, and vice versa.

It is not clear to what extent a TSMS should be able to cope with these data or how a TSMS should be combined with another DBMS to manage them.

Time Series Bases

All the time series, groups and other data managed as one logical unit make up a time series base. Its size depends on the number of time series, their periodicity, the observation interval, and the size of the events.

2.1.2 Functional Requirements

Distribution of Functional Capabilities

As mentioned before, researchers use their time series with several programs, such as statistics packages, spreadsheets, and charting programs. These specialized programs provide a thorough functionality in their application area. Therefore, it makes no sense for the TSMS to replicate all the functionality these programs already offer. It is more sensible to build an environment where the TSMS works as a repository for the time series, and the tools use this repository to provide their special services. The emphasis of the TSMS should, therefore, be on data management capabilities and on basic transformation capabilities to prepare the data for further analysis.

Operations on Events

A TSMS should support the usual operations on atomic types, and must provide read and update access to the components of structured event types.

Operations on Time Series

The most important functional capabilities of a TSMS are the definition of new time series, the storing of data in time series, and the retrieval of data from time series. Another important capability of a TSMS is the derivation of new time series from existing ones. Examples are the computation of the difference of two time series, the application of a moving average or other filters to a time series, and the transformation of the periodicity of a time series.

A TSMS should provide decent query capabilities. It should at least support a select operation for time series similar to the select operation of the relational algebra. It would also be useful to be able to apply not only logical operations in the selection condition but also other operations, such as arithmetic and time-related operations. An optimizer should support users in query processing to achieve a high performance level.

Operations on Groups

Additional operations are required to define groups, to store and to retrieve group data, to add and remove members, and to enumerate them. The manipulation of the member set should be facilitated by set operations. One should also be able to apply operations to all members of a group without explicitly iterating over them.

Calendar

A TSMS must support calendar-related operations and date arithmetic, e.g. computing the difference between two dates, even if the two time series have different periodicities. Different time series may be based on different calendars, e.g. a business calendar with five business days per week or a calendar that can cope with local holidays.

Browsing and Editing

Two other important functions are browsing and editing. Browsing means interactively exploring the time series available in a time series base and investigating their contents. Such a browsing tool also helps to get an overview of the group structures. Editing means interactively building new or updating existing time series.

A browser-editor also needs some presentation facilities capable of displaying time series in table or chart formats. However, these presentation facilities are rather basic. For a more sophisticated presentation, one can use specialized tools.

So far, we have described requirements of a TSMS data model. One of the goals of the project will be defining a data model and deciding which of the structures and functional capabilities mentioned should be incorporated.

2.2 Data Exchange Requirements

2.2.1 Import of Data from Various Data Sources

An important issue in system architecture concerns the distribution of databases. Researchers have a private time series base for their individual research, but they also need access to other databases, e.g. time series bases of other members within the research group, or company-wide financial databases. Most of these databases do not use the same data model or the same schema as the private time series base. They may reside on the same computer, on the same local area network or on a remote computer.

Researchers frequently want to copy a subset of such other databases into their private time series base. Such a copy operation may be triggered manually, by some time event or by some other event of interest, e.g. a price crossing a certain threshold. Therefore, specifying various update policies and events in a TSMS must be easy.

In addition to databases, there are various other data sources. Data may be gathered manually. At UBS, for example, the Gross National Product of several countries is estimated and entered manually. Other data are received as files from external data providers like OECD. Furthermore, data is also received as a continuous data stream from other computers. Stock prices, for example, can be received as a real time data feed from ticker services such as Reuters or Telekurs.

Data sources differ in the amount of data and in the frequency of updates. Manual entry generates only a small amount of new data and the frequency of updates is low. In contrast, a real time data feed often generates a large amount of data. However, for research purposes, it is normally sufficient if the TSMS can cope with fewer update requests at regular intervals.

Usually, the format of data from various sources also differs. A TSMS must be able to handle all these formats.

Normally, the available file formats for data exchange are not satisfying, because they do not define how to format header data like periodicity, etc. It might be necessary to develop a specific format to make lossless data exchange possible.

2.2.2 Export of Data to Client Applications

When we export data to programs like statistical packages or spreadsheets, we are faced with similar problems as with data import.

2.2.3 Means of Data Exchange

The different sources for data import and sinks for data export provide different ways of data exchange. Interfaces between components can be realized, for example, via program-to-program communication or simple file transfer. The choice of the appropriate solution depends on the connection facilities: e.g. there may be a distributed environment that allows RPCs, a file transfer protocol like FTP, or just a connection via electronic mail. Therefore, a TSMS should support different ways of data exchange.

2.3 Data Quality Management Requirements

Time series management makes high demands on data quality management. New raw data has to be checked for consistency with older events, outliers have to be detected, noise has to be filtered to clean up the data, etc. To track the value of calibrated data back to the raw data, it is often necessary to retain the raw and the calibrated data.

Old estimations have to be replaced by newer ones when new information is available, but one might like to retain the older data to review and improve the estimation process. This requires a versioning concept.

It should also be possible to store information on the quality of the data, whether for the entire time series or just for individual events.

2.4 Synchronization Requirements

Usually, there is only one process writing a time series while all other processes just read it. However, it is not yet clear whether it is enough to support only a single writer / multiple reader transaction model.

Ordinarily, new data is appended to the end of a time series and all other events are rarely modified. This might facilitate the synchronization of multiple users.

3 Example of a Time Series Base

An example of a time series base is HIKU (Historische Kurse, i.e. historical rates) from the Swiss company Telekurs. It contains approximately 10'000 time series from the financial world. The system has been operational since 1992, and the time series run from 1986 (1983 for Swiss shares). The quality of the data is checked by a specialized company in Germany.

The time series are updated daily. Data are compressed and transferred to customers via file transfer over X.25. Telekurs offers software for workstations and PCs which handles the communication and the compression/decompression of the data. The size of the whole time series base is approximately 4 GByte.

There are three kinds of time series in HIKU: Prices of financial instruments with raw values, prices of financial instruments with adjusted values, and adjustment factor time series. All their headers and events have lengths from 18 to 44 Byte. The first two kinds of time series have basically the same format. The header contains data like the number of the financial instrument, the type (e.g. stock or share), the currency, the location of the stock exchange, the branch code, and the number of events. The time series with raw values contain nominal prices; they do not consider changes in prices resulting from splits of shares, the emission of bonus shares, etc. The time series with adjusted values take these changes into account. An event contains, among others: the date, different prices (closing, high, and low; opening prices are not yet available), trading volume, etc. The header of an adjustment factor time series contains a subset of the header described above. The events contain the date and the adjustment factor.

Time series are selected according to standard selection files which offer some predefined selection criteria, such as Blue Chips International, European Stock Exchange, Swiss Indices, and Foreign Exchange Rates. These criteria may serve to define groups. However, users cannot select groups according to their own criteria, e.g. all bank shares.

4 Current Solutions and Related Research Work

Time series management with files

When time series are stored in simple files, there are several drawbacks: The advantages of a DBMS get lost. The concepts of groups and time series bases are not really supported. Functional capabilities have to be implemented as separate programs on top of the file system. The same is true for data exchange as well as for data quality management.

Relational DBMS

Time series management with RDBMS also has considerable disadvantages, e.g. time series are based on sequences, whereas the relational data model uses a set concept. An RDBMS is neither suited to model recursive structures nor to handle heterogeneous sets. The time concept within current RDBMS is not very sophisticated.

This means that data processing specialists would have to write specific applications for the economic researchers on top of the RDBMS.

Object-oriented DBMS

OODBMS [Catt 91] provide a lot more capabilities to model and implement time series than RDBMS. The main advantage is that arbitrary data types can be modeled as classes. Information hiding, inheritance, and reusability are further strengths of OODBMS. Very useful concepts for time series management are the declaration of data types such as ordered collections or sets and the handling of recursive structures and heterogeneous sets. A sophisticated time concept, including calendar functionality, can be modeled as special classes, and methods can realize complex operations on time series. Therefore, an OODBMS is a good basis on top of which a TSMS can be built.

Specialized TSMS

To our knowledge, there are currently only very few commercial DBMSs specialized for time series management. The most mature and interesting of them is the *FAME* system [Kotz 92]. FAME has many useful features, but also disadvantages, e.g. poor search and retrieval facilities, and no mechanisms for data quality management and data consistency control. The data model is not powerful enough: each event may have only one scalar field, and the group concept – lists of time series names from which members may be selected by pattern matching with simple wild-cards, not by content – is too limited. Finally, the 4GL requires a lot of experience and special training.

Related research work

A lot of work has been done in the field of *temporal databases* ([SA 86], [Tans 93], [SS 92]). However, most temporal database systems use an interval model, whereas a TSMS needs a time point model.

The temporal data model closest to our requirements is described in [SS 87], [SS 93], [CS 93] and [SC 93]. In the former version of the model, the main constructs are time sequences (TS) and time sequence collections (TSC). Though being central modeling concepts, TSs and TSCs are treated as weak entities only. As a TSC includes the time series of all objects of a class, a time series cannot be directly accessed but has always to be selected from the TSC. TSCs must be homogeneous regarding all properties, i.e. the properties of time sequences cannot vary within the same collection (e.g. to record economic values of two countries at different granularity). Furthermore, no means are provided to arbitrarily group time series from different classes.

In the newer version of the model, these characteristics do no longer apply. However, as the model is based on a predefined set of extended relational operations, its expressional power with respect to time series transformation, filtering etc. is limited. For example, the model in [SC 93] does provide an interpolation function for time series (the "type" of a time series), but neither an aggregation function nor individual interpolation functions for different attributes of the same series. The notion of "Concept" found in [SC 93] mixes three orthogonal ideas: inheritance (IS-A hierarchy of Concepts), grouping of time series according to some common usage feature and a view mechanism based on event construction.

Statistical databases [Mich 91], [HF 92], *scientific databases* [Mich 91], [HF 92], and *spatial databases* [LT 92] address certain problems that a TSMS has to deal with, too. However, they give no preference to the time dimension, as a TSMS is supposed to do.

5 Goals for a Research Project in Time Series Management

The current state of the art mentioned above led us to the conclusion that it is worthwhile to start a project on time series management with the following goals:

- The TSMS must be suited to end users.
- The data model must support managing many time series with record-like events and partitioning the time series of a time series base into various groups.
- The system has to support the derivation of new time series and groups from existing ones.
- Retrieval of time series and groups must be provided through a general search mechanism and adequate browser tools.
- Facilities for efficient and easy to use data quality management have to be provided.
- The system must be able to handle data exchange between a collection of loosely coupled time series bases, a variety of other data sources, and data sinks.
- The TSMS must allow group use.

In the following, we will discuss some of these goals.

A system suited to end users

End users require simple basic concepts (e.g. with respect to the data model) and user-friendly manipulation facilities (e.g. a graphical user interface consisting of several browsers and editors, as described in chapter 2.1.2).

Data model

We propose a data model that comprises three basic elements: events, time series, and groups. These elements are explained in chapter 2.1.1.

Our TSMS will not only allow time series and groups to be stored and retrieved, but it will also support the transformation of time series and the derivation of new time series.

Retrieval of time series and groups

In addition to the browsers explained in chapter 2.1.2, more traditional query facilities allow users to search for groups and time series based on their data contents.

Data exchange

The different components and the conditions under which data exchange takes place have been described in chapter 2.2.

Our intent is to build a generic specification formalism that describes all the parameters necessary for interoperability, e.g. the data formats, the source and destination of the data, means of data exchange and exchange frequency. The data exchange will be set up automatically according to the specification [Drey 93].

A system for personal and group use

Concurrent updates to time series will be synchronized by a concurrency control mechanism. As stated in chapter 2.4, such a mechanism can probably be realized with a rather simple approach, because there is normally just one writer per time series at a time.

6 Research Issues

The current situation and the goals we set for our project result in a number of research challenges:

- *Definition of a time series data model:* The data model must have enough expressive power to model a broad spectrum of time series types and group types. It must incorporate the functional capabilities to support the derivation of new time series, the filtering of time series, etc.
- *Implementation of a specialized DBMS on top of an off-the-shelf OODBMS:* We intend to build the TSMS on top of a commercially available, unmodified OODBMS. It will be interesting to experience in what respect the OODBMS facilitates the construction and which features are missing.
- *Data exchange concept:* There are many discussions going on in the area of database interoperability. These discussions concern topics like data replication or schema integration. However, the question of how to best handle multiple private and public databases which loosely cooperate via data exchange has not yet been settled.
- *Queries over heterogeneous collections:* In our TSMS, a group can contain various types of time series and groups. The realization of a flexible yet efficient query facility that can handle such heterogeneous sets is an interesting problem.
- *Data quality management:* Some conceptual work has been done in the area of data quality management [Wang 93], but few systems seamlessly integrate data quality concepts. The existing solutions are rather ad hoc than systematic.

7 Conclusions

Time series management raises a variety of questions ranging from data structures and functional capabilities to user interfaces and interoperability. Existing solutions are not satisfying, and related research work is only partially directed towards our problem domain.

It is for these reasons that a research project in time series management seems to be a worthwhile undertaking. The system's major goals will lie in the fields of retrieval and transformation capabilities, end user orientation, data exchange and data quality management. We also hope to gain insight into the building of special-purpose data management systems, which could ease the development of future systems.

References

[Catt 91] R.G.G. Cattell: Object Data Management - Object-Oriented and Extended Relational Database Systems. Addison-Wesley, 1991.

[CS 93] R.Chandra, A.Segev: Managing Temporal Financial Data in an Extensible Database. Proc. of the 19th VLDB Conf., Dublin 1993.

[Drey 93] W. Dreyer: Interoperability Issues in Time Series Management. Proceedings of the Workshop on Interoperability of Database Systems and Database Applications; Schweizer Informatiker Gesellschaft, Data Bases - Theory and Application; to be published in fall 1993.

[HF 92] H. Hinterberger, J.C. French (eds.): Proc. 6th Intl. Working Conference on Scientific and Statistical Database Management. Ascona, 1992.

[Kotz 92] A. Kotz Dittrich: FAME - A Data Management System for Time Series. UBILAB Report 92.02.28, 1992 (in german).

[LT 92] R. Laurini, D. Thompson: Fundamentals of Spatial Information Systems. Academic Press, 1992.

[Mich 91] Z. Michalewicz: Statistical and Scientific Databases. Ellis Horwood Ltd., 1991.

[SA 86] R.T. Snodgrass, I. Ahn: Temporal Databases. IEEE Computer, 19 (9), Sept. 1986.

[SC 93] A. Segev, R. Chandra: A Data Model for Time-Series Analysis. Workshop on Current Issues in Databases and Applications, Rutgers Univ., Oct 1992. In: Advanced Database Systems, editors: N. Adam and B. Bhargava, Lectures Notes in Computer Science Series, Springer Verlag, 1993.

[SS 87] A. Segev, A. Shoshani: Logical Modeling of Temporal Data. In Proc. of the ACM SIGMOD Annual Conf. on Management of Data, May 1987.

[SS 92] M.D. Soo, R.T. Snodgrass: Multiple Calendar Support for Conventional Database Management Systems. Univ. of Arizona, Dept. of Comp. Science, TR 92-07, Feb. 1992.

[SS 93] A. Segev, A. Shoshani: A Temporal Data Model Based on Time Sequences. In [Tans 93], chapter 11, pp. 248 - 269.

[Tans 93] A.U. Tansel et al.: Temporal Databases - Theory, Design and Implementation. The Benjamin/Cummings Publ. Comp., 1993.

[Wang 93] R. Y. Wang et al.: Data Quality Requirements Analysis and Modeling. In Proc. of the 9th Int. Conf. on Data Engineering, April 1993.