Data warehouses-TOLAP-decision making.

Panagiotis Chountas[1]
Christos Vasilakis[1]
Elia El-Darzi[1]
Ilias Petrounias[2]
Andy Tseng[2]

[1] Harrow School of Computer Science, University of Westminster
[2] Department of Computation, UMIST

Data Warehouses-TOLAP-Decision Making*

P. Chountas, C. Vasilakis, E. El-Darzi
School of Computer Science, University of Westminster Watford Road, Northwick Park, London, HA1 3TP, UK
chountp@wmin.ac.uk

I. Petrounias, A. Tseng
Department of Computation, University of Manchester Institute of Science & Technology, PO Box 88, Manchester M60 1QD, UK
ilias@co.umist.ac.uk

Abstract - Data warehouses (DWH) have been established as the core of decision support systems. On top of a DWH, different applications can be realised with regard to conventional reporting. On line Analytical Processing (OLAP) has reached the maturity as an interactive and explorative way of analysing DWH data. However DWH are mostly organised as snapshot databases. For this reason important tasks like "how many times have products of a specific brand been sold in the "past?" cannot be answered successfully - in order to control the success of reshuffling the product range it is necessary to compare the sales of "old" and "new" products. The same applies in cases where the seasonality aspect for a particular range of products has to be answered. On the other hand, temporal databases allow a valid time to be assigned to data. In this manner, a past state can be reconstructed during retrieval. In this paper, we address the integration of DWH and OLAP with *temporal database semantics*.

Keywords: Temporality, Time, Uncertainty, Temporal Data warehouses.

1 Introduction

This paper concentrates on the temporal aspects of data warehouses and their effects on OLAP environments. We suggest a temporal model for multidimensional DWH-OLAP, motivated by the observation that ignoring temporal issues leads to questionable expressive power and query semantics in many real life scenarios. Our suggested model will allow the expression of temporal OLAP queries in an elegant and intuitive fashion.

We introduce multidimensional modelling for demonstrating the conventional OLAP architecture, and introduce the term temporal OLAP, TOLAP. A TOLAP environment is an extended OLAP environment that is able to handle temporal data and semantics. In the light of the above, we introduce a temporal multidimensional data model and a temporal SQL-type query language named as TOQL. We introduce TOLAP by means of examples, and we formally define its syntax and semantics. We propose the extension of the existing DWH/OLAP environment by incorporating temporal aspects. Therefore we require valid time information about data as well as the ability to model changing dimensional data and hierarchies. The OLAP query mechanism is extended to be able to execute temporal queries. In doing this we are proposing a query language that is extended with <Time> ON <DIMENSION> clause. The proposed extension paves the way towards TOQL query formalism for temporal OLAP environments. Furthermore, we suggest a new operator as part of TOQL formalism for dealing with multidimensional information that does not exist at a specific valid time.

The rest of the paper is organized as follows: in section two we define the impact of time in the architecture of a data warehouse. Section three delivers a time model for defining evolving hierarchies with either implicit or explicit temporal semantics. The representation of temporal data as part of a TOLAP environment is defined in section four with the emphasis in delivering a query model based on similarity empowerment for defining OLAP operators over changing hierarchies. Finally we conclude and provide an outlook on future research.

2 Temporality & Data warehouses

Temporal data warehouses should describe the evolving history of an enterprise. In the case of patient record data, it is frequently very important to enable *the monitoring* of data changes, i.e. to retain a complete history of past states. Correcting errors could be possible by posting compensating transactions with different timestamps to the data warehouse. In health informatics applications, keeping track of the diagnosis on which decisions where made may guard against wrongful, misconduct claims. When considering temporal DWH's we need to understand how time is mirrored in a temporal database and how this relates to the structure of the data. Temporal DWH's usually have to accommodate the following type of data;

Regular data. Once a record is added to a database, it is never physically deleted, nor is its content ever modified. Rather, new records are always added to reflect transactions on data. Regular data thus offers a complete

* 0-7803-7952-7/03/$17.00 © 2003 IEEE

history of the changes occurred in the data. Temporal DWH's contain regular data

Snapshot data. A data snapshot is a stable view of data as it exists at some point in time. It is a special kind of regular data. Snapshots usually represent the data at some time in the past, and a series of snapshots can provide a view of the history of an enterprise.

The focus of existing research [2] in temporal data warehouses is on storing "regular" data with the aid of time stamped status and event records. The intuition behind this stream of research is that a query needs to access current data. In a single timestamp scheme, the only way to identify current records is to find the latest timestamp of the regular set, which is an inefficient process. It has been proposed [1] time to be treated as a dimension and also to be considered as an intrinsic element of the fact table, see Figure 1.

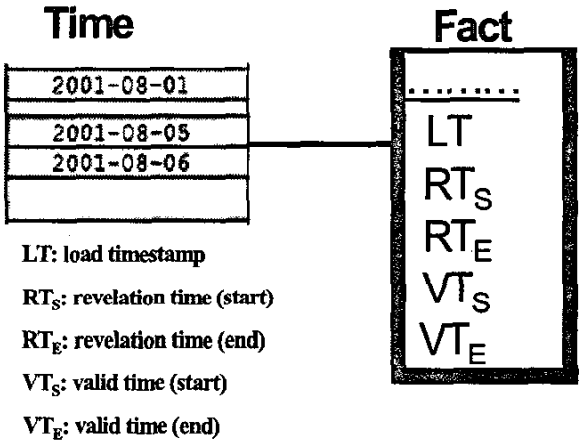


Figure 1. Temporal Fact Table

Eventually what is proposed by is an extension of the bi-temporal database model [4], with the inclusion of the load timestamp. The load timestamp is basically addressing the need of knowing when a piece of information was integrated in the data warehouse, while the revelation time is indicating when a piece of information was recorded as part of a particular source.

This paper is focused on the impact of the valid time dimension in multidimensional analysis. More specifically we advocate that:

- The assortment of a particular hierarchy may be variable through the valid time dimension. For example considering the disease assortment, a new disease may appear, or a disease may move from one group to another.
- It may be possible to know for how long a piece of information is valid i.e. the length of the time interval is known, though the starting or the ending point may not be defined with precision. To put it differently the time dimension itself may be evolving.

In the next sections the temporal concepts that will allow the development of a temporal data warehouse as well as the formulation of TOLAP queries are defined.

3 Modelling Valid Time

Time has a standard geometric metaphor. In this metaphor, time itself is a line; a point on the time-line is called a *time point*; and the time between two *time points* is known as a *time interval.*

Time point: Our model of time does not mandate a specific time point size or *(minimum) granularity*; a time point may be of any duration (e.g., nanoseconds, years, Chinese imperial dynasties). We believe that specifying the minimum granularity should be left to the implementation rather than be fixed in the data model. Although our time model has only a single granularity, multiple granularities can also be handled. The time points are consecutively labeled with the integers in the set T = *{0,...,N}* where *N* is the number of different values that a timestamp can represent. The set of time points is linearly ordered.

Time Line: is the geometric metaphor of the time. Conceptually, time is linear and consists of a set of time points. The time line is represented with the aid of the linear equation:

$$KX + B: (K, B, X) \in \mathrm{N} \wedge C \leq \mathrm{B} \leq C', (\mathrm{C}, C') \in \mathrm{N} \qquad (1)$$

Time Interval: A time interval is an instantiation of a time line and is bounded between two-time points of a specific duration. For example, assume the following triple values for the variables *(K, B, X)* respectively *(K=0, X=0, 3 ≤ B ≤ 5)*. This will generate the time interval $T_I=[t_L, t_R]=[3,5]$ that may imply any particular interval and can be mapped to any time hierarchy or calendar.

Time Hierarchy-Calendar: A *linear hierarchy of time units*, denoted H_r, is a finite collection of distinct time units with a linear order $\sqsubseteq$ among those time units. H_r is a finite collection of distinct time units, with linear order among those units, e.g. H_1=day⊑month⊑year, H_2=⊑hour⊑day⊑month⊑year are linear hierarchies of time units defined over the Gregorian calendar. Thus, a calendar is a collection of linear time hierarchies. A *calendar* consists of a linear hierarchy *H* of time units and a *validity predicate* denoted *valid*H_r. A validity predicate specifies a non-empty set of valid time points; *valid*H_r (*t*) is true if *t* is a valid time point. A validity predicate states that, for example, valid (14/9/1995) = true but valid (29/2/1997) = false. However (30/2/1997) is not a valid time point since February of 1997 only contains 28 days. Conceptually, time may be extended to infinite (⊥,⊤) past or future.

Duration(*I*): is the length of a time interval. To prevent having ill-formed temporal intervals the specified

length is not the distance between the n and n-1 projections over the set of time points T.

The following functions define formally the starting point of a time interval $\vdash(t)$, the ending time point of a time interval $\dashv t)$ and its length with respect to an arbitrary calendar$|\cdot|(t)$.

$$\vdash(C, C', l) \rightarrow C \quad (2)$$

$$\dashv(C, C', l) \rightarrow C' \quad (3)$$

$$|\cdot|(C, C', l) \rightarrow l \quad (4)$$

It is now possible to extend the set of temporal intervals with the inclusion of null values {?}. To make the three functions given above work correctly we must extend them to include null values ($_?$):

$$\vdash(C, C', l)_? \rightarrow \{C, \vee (C'-l)\} \quad (5)$$

$$\dashv(C, C', l)_? \rightarrow \{C' \vee (C+l)\} \quad (6)$$

$$|\cdot|(C, C', l)_? \rightarrow \{l \vee (C'-C)\} \quad (7)$$

The proposed time representation for the valid time dimension can be used for encoding two different types [3] of temporal information:

Definite Temporal Information: is defined over the interval $T_1=[C, C']$ when the time points C, C' are defined with precision over the time line. Using our time model definite temporal information can be utilised with the aid of equations (2), (3), (4).

Indefinite Temporal Information: is defined over the interval $T_1=[C, C']_?$ when the time points C, C' are not defined with precision over the time line but are bounded. Thus the time dimension itself is evolving. Indefinite temporal information in the context of our time reorientation can be utilised with the aid of equations (5), (6), (7).

Furthermore, the inclusion of "null" in the set of time intervals allow us to encode implicit temporal information that cannot be represented with the use of an explicit–strict temporal representation. Moreover, the proposed time model does not treat time as a way to index propositions or information [5], [6] since the time itself is treated as a dynamic property.

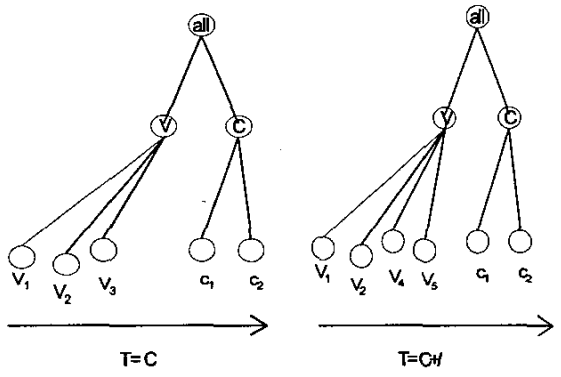
The last observation poses the question how OLAP analysis can be performed as part of a temporal environment where:

a) Hierarchy trees are changing through different times
b) The valid time for requesting OLAP to be performed over a hierarchy tree may be implicit.

4 Temporal Data & TOLAP

Current approaches [7,8] assume that cube facts have an implicit timestamp assigned by their time dimension. In contrast, the dimensional elements are considered as snapshots. However, this kind of treatment does not take into account the fact that hierarchical structures can change over time. In this case, a typical requirement analysis could be to compare the new grouping with the old one.

In solving this problem, we can annotate the edges of a hierarchy tree with valid time intervals. In this matrix, the parent nodes are the rows while children nodes are considered as columns; every cell takes a set of valid time intervals. This meta-information can be defined for every hierarchy tree in the data warehouse schema. Figure 2 presents the hierarchy tree of an arbitrary hierarchy tree for the arbitrary time interval [C, (C+*l*)] where {C, (C+*l*)}$\in H_r(t)$. The corresponding valid matrix is also presented. For illustrative purpose we assume that $\{V_1, V_2, V_3, V_4, V_5, C_1, C_2\} \in N$.



	V_1	V_2	V_3	V_4	V_5	C_1	C_2	V	C
V	⊥ T	⊥T	⊥,C	C+l , T	C+l, T				
C						⊥T	⊥ T		
All								⊥ T	⊥ T

Figure 2. Hierarchy trees & corresponding valid time matrix

The assumption is that the defined hierarchy trees and valid time matrix depict a possible temporal analysis trace. Such analysis can be performed with the aid of a TOLAP query. The formulation of such a query requires the setting of a valid time for each dimension. Therefore, a query language can be extended by a "WITH TIME INTERVAL <Hr> On DIMENSION <DIMENSION>" clause. For each dimension the user is required to select an explicit or implicit time interval-valid time.

Such a query is capable of defining for example, the ROLL-UP effect with respect to distinct time points C, (C+*l*) as well as with respect the time interval [C, (C+*l*)].

4.1 The TOLAP Query Model

We put forward a proposal to provide representation for handling changing-evolving hierarchies and retrieving intentional answers at the query level through case based reasoning mechanisms (CBR).

The motivation behind this extension is based on the argument that in order to build TOLAP queries with their own information bases, out of the existing ones, we need to raise the abstraction of operations on the metadata level. In this context, the case-based querying-technology can be focused on an alternative mechanism for designing intelligent TOLAP querying systems. It can be employed either at the conceptual level or at the instance level (metadata level) for matching a current query description (a query case) to a specially organised database of indexed previous situations, called a case base (info-base). Therefore, a TOLAP querying system searches for case histories (response cases) that fully or partially match this description. The CBR strong features spring from its emphasis on similarity matching.

Empowered similarity requires a model that elegantly combines into a sole formula both hierarchy similarities and object dissimilarities with respect to distinct time points C, (C+*l*). A simple model to capture similarities and dissimilarities between objects was proposed by [9], could be summarised as follows:

$$sim(H_A, H_B) = S(H_A \cap H_B) - S(H_A - H_B) - S(H_B - H_A) \quad (8).$$

In the context of a temporal environment this can be interpreted as follows: given two sample hierarchy trees H_A, H_B defined in the time interval [C, (C+*l*)], estimate the ROLL-UP effect with respect to distinct time points C, (C+*l*) as well as for the time interval [C, (C+*l*)].

H_{AB} should contain all Trevskys components, ($S(H_A \cap H_B)$, $S(H_A - H_B)$, $S(H_B - H_A)$), respectively. The intuition is that the temporal ROLL-UP result, must express equally well the similarities ($S(H_A \cap H_B)$) and dissimilarities ($S(H_A - H_B)$, $S(H_B - H_A)$) between the evolving hierarchies (H_A, H_B).

The important issue in constructing Trevsky's components is the estimation of the $S(H_A \cap H_B) - S(H_A - H_B) - S(H_B - H_A)$ parameters for the distinct time interval [C, (C+*l*)].

With reference to Figure-2 hierarchies Trevsky's components for the time interval [C, (C+*l*)], are estimated as follows:

$$S(H_A \cap H_B)_{[C,(C+l)]} = \{\{V_1, V_2, \}\{C_1, C_2\}\} \text{ items} \quad (9)$$

$$S(H_A - H_B)_{[C,(C+l)]} = \{\{V_3\}\} \text{ items} \quad (10)$$

$$S(H_B - H_A)_{[C,(C+l)]} = \{\{V_4, V_5, \}\} \text{ items} \quad (11)$$

With reference to the distinct time points C or (C+*l*) an extended ROLL-UP operator should reflect the effects occur by the time query:

- Items not valid at a particular time point
- Groupings with same name but different elements
- Residual items excluded from the chosen time point

While drilling down can solve the first and second issues, the last issue requires an explanation of the instances hidden below the grouping "Residual".

The OLAP architecture has to be modified as follows: information about valid time has to be stored in the meta-data repository and the OLAP server must be able to receive queries with valid time clauses. The repository itself has to be extended for storing versioned meta-information.

5 Conclusions

Traditionally, there is no real-time connection between a DWH and its data sources. This is mainly because the write-once read-many decision support characteristics would conflict with the continuous update workload of operational systems resulting in poor response times. Consequently, up until recently, *timeliness* requirements were restricted to mid-term or long-term time windows. Ignoring these temporal issues leads to diminished expressive flexibility and questionable query semantics in many real-life scenarios.

We review the issue of time in data warehouses and meaning of time as part of a conventional OLAP architecture. We propose an extension of the conventional OLAP architecture in order to handle temporal data. The extension concerns metadata repository while the data warehouse remains untouched.

In enforcing the TOLAP architecture as a software component the following issues have to be tackled:

- The communication between the OLAP server and the metadata repository has to be defined.
- Efficient algorithms for the proposed schema evolution have to be considered.
- An indication of whether information is aggregated or whether data have been transformed into another dimensional structure.

The advantages provided by built-in temporal consistency support in data warehouses include: higher reliability in data modeling, more efficient gathering of an organization's history, as well as analyzing the sequence of changes to that history.

Important future research directions in this field will be the maintenance of data warehouses based on dynamically changing information systems (data updates, schema changes), and enhancements to the active behaviour in the field of active data warehouses.

We further suggest the analysis of the process dimension with the capturing of case items or contextual facts as part of an integrated temporal workflow environment. The functionality of such environment is to express changing case-items with the aid of moving hierarchies-dimensions.

References

[1] S. Anahory, Murray D. Data warehousing in the real world: A practical approach for building Decision Support Systems, Addison-Wesley 1997

[2] M.H. Böhlen et al. Point- versus Interval-Based Temporal Data Models. In Proc. of 14th ICDE, IEEE Computer Society Press, pp. 192-201, Orlando, Florida, USA, 1998.

[3] P. Chountas, I. Petrounias, "Representation of Definite, Indefinite and Infinite Temporal Information", Proc. 4th Int. Database Engineering & Applications Sym., IEEE Computer Society Press, 2000, pp 167-178, 2000

[4] O. Etzion, S. Jajodia, and S. Sripada, (editors). Temporal Databases: Research and Practice, LNCS, 1998.

[5] W.H. Inmon. Building the Data Warehouse. Second Edition, John Wiley & Sons, New York, 1996.

[6] C.S. Jensen and R.T. Snodgrass. Temporal Data Management. In IEEE Transactions on Knowledge and Data Engineering, Vol. 11(1): pp. 36-44, 1999.

[7] R. Kimball. The Data Warehouse Toolkit Jon Wiley & Sons, New York, 1996.

[8] T.B.Pedersen and C.S. Jensen. Multidimensional Data Modeling for Complex Data. In Proc. of 15th ICDE, IEEE Computer Society, pp. 336-345, Sydney, Australia, 1999.

[9] A. Trevsky, I.Gati, "Studies of similarity", Cognition and Categorization, Hillsdale, NJ: Erlbaum, 1978