Temporal Extension for a Conceptual Multidimensional Model

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

Data warehouses integrate data from different source systems to support the decision process of users at different management levels. Data warehouses rely on a multidimensional view of data usually represented as relational tables with structures called *star* or *snowflake schemas*. These consist of *fact tables*, which link to other relations called *dimension tables*. A fact table represents the focus of analysis (e.g., analysis of sales) and typically includes attributes called *measures*. Measures are usually numeric values (e.g., quantity) used for performing quantitative evaluation of different aspects in an organization. Measures can be analyzed according to different analysis criteria or *dimensions* (e.g., store dimension). Dimensions may include *hierarchies* (e.g., month-year in the time dimension) for analyzing measures at different levels of detail. This analysis can be done using on-line analytical processing (OLAP) systems, which allow dynamic data manipulations and aggregations. For example, the roll-up operation transforms detailed measures into aggregated data (e.g., daily into monthly or yearly sales) while the drill-down operations does the contrary.

Multidimensional models include a time dimension indicating the timeframe for measures, e.g., 100 units of a product were sold in March 2007. However, the time dimension cannot be used to keep track of changes in other dimensions, e.g., when a product changes its ingredients. In many cases the changes of dimension data and the time when they have occurred are important for analysis purposes. Kimball and Ross (2002) proposed several implementation solutions for this problem in the context of relational databases, the so-called *slowly-changing dimensions.* Nevertheless, these solutions are not satisfactory since either they do not preserve the entire history of data or are difficult to implement. Further, they do not consider the research realized in the field of temporal databases.

Temporal databases are databases that support some aspects of time (Jensen & Snodgrass, 2000). This support is provided by means of different temporality types[1], to which we refer in the next section. However, even though temporal databases allow to represent and to manage time-varying information, they do not provide facilities for supporting decision-making process when aggregations of high volumes of historical data are required. Therefore, a new field called *temporal data warehouses* joins the research achievements of temporal databases and data warehouses in order to manage time-varying multidimensional data.

BACKGROUND

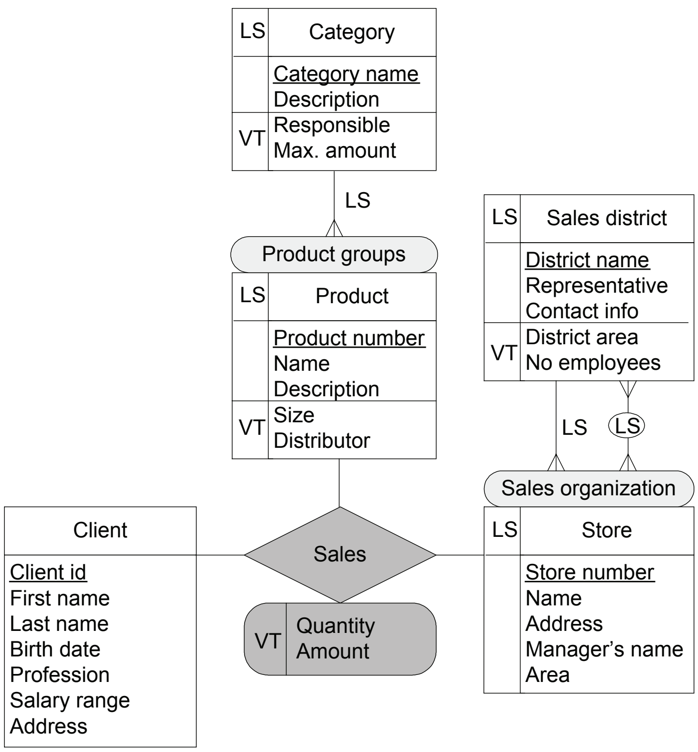
Temporal support in data warehouses is based on the different temporality types used in temporal databases. *Valid time* (VT) specifies the time when data is true in the modeled reality, e.g., the time when a specific salary was paid for an employee. Valid time is typically provided by users. *Transaction time* (TT) indicates the time when data is current in the database and may be retrieved. It is generated by the database management system (DBMS). Both temporality types, i.e., valid time and transaction time, can be combined defining *bitemporal time* (BT). Finally, changes in time defined for an object as a whole define the *lifespan* (LS) of an object, e.g., the time when an employee was working for a company.

One characteristic of temporality types is their precision or *granularity*, indicating the duration of the time units that are relevant for an application. For example, the valid time associated to an employee's salary may be of granularity month. On the other hand, transaction time being system defined is typically of a millisecond granularity.

There is still lack of an analysis determining which temporal support is important for data warehouse applications. Most works consider valid time (e.g., Body, Miquel, Bédard, & Tchounikine, 2003; Wrembel & Bebel, 2007; Mendelzon & Vaisman, 2003). To our knowledge, no work includes lifespan support in temporal data warehouses. However, lifespan is important since it can help to discover, e.g., how the exclusion of some products influences sales. On the other hand, very few works relate to transaction time. For example, Martín and Abelló (2003) transform transaction time from source systems to represent valid time. This approach is semantically incorrect because data may be included in databases after their period of validity has expired. Further, transaction time coming from source system plays an important role in temporal data warehouses when traceability is required, e.g., for fraud detection. Other authors consider transaction time generated in temporal data warehouses in the same way as transaction time in temporal databases (e.g., Martín & Abelló, 2003; Mendelzon & Vaisman, 2003; Ravat & Teste, 2006). However, since data in temporal data warehouses is neither modified nor deleted, transaction time in a data warehouse represents the time when data was loaded into a data warehouse. Therefore, we propose to call it *loading time* (LT) (Malinowski & Zimányi, 2006a). LT can differ from transaction time or valid time of source systems due to the delay between the time when the changes have occurred in source systems and the time when these changes are integrated into a temporal data warehouse. Another approach (Brucker & Tjoa, 2002) considers valid time, transaction time, and loading time. However, they limit the usefulness of these temporality types for only active data warehouses, i.e., for data warehouses that include event-condition-action rules (or triggers).

The inclusion of temporal support raise many issues, such as efficient temporal aggregation of multidimensional data (Moon, Vega, & Immanuel, 2003), correct aggregation in presence of data and schema changes (Body *et al.*, 2003; Eder, Koncilia, & Morzy, 2002; Wrembel & Bebel, 2007; Mendelzon & Vaisman, 2003; Golfarelli, Lechtenbörger, Rizzi, & Vossen, 2006), or temporal view materialization from non-temporal sources (Yang & Widom, 1998). Even though the works related to schema and data changes define models for temporal data warehouses, what is still missing is a conceptual model that allows decision-making

Figure 1. An example of a multidimensional schema for a temporal data warehouse



Related Content

Fuzzy Methods in Data Mining
Eyke Hüllermeier (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 907-912).*
www.irma-international.org/chapter/fuzzy-methods-data-mining/10928

Data Mining Applications in the Hospitality Industry
Soo Kim (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 406-410).*
www.irma-international.org/chapter/data-mining-applications-hospitality-industry/10852

Interest Pixel Mining
Qi Li, Jieping Yeand Chandra Kambhamettu (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 1091-1096).*
www.irma-international.org/chapter/interest-pixel-mining/10957

Quantization of Continuous Data for Pattern Based Rule Extraction
Andrew Hamilton-Wrightand Daniel W. Stashuk (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 1646-1652).*
www.irma-international.org/chapter/quantization-continuous-data-pattern-based/11039

Deep Web Mining through Web Services
Monica Maceliand Min Song (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 631-637).*
www.irma-international.org/chapter/deep-web-mining-through-web/10887