Chapter 2.16
Managing Late Measurements in Data Warehouses

Matteo Golfarelli
University of Bologna, Italy

Stefano Rizzi
University of Bologna, Italy

ABSTRACT

Though in most data warehousing applications no relevance is given to the time when events are recorded, some domains call for a different behavior. In particular, whenever late measurements of events take place, and particularly when the events registered are subject to further updates, the traditional design solutions fail in preserving accountability and query consistency. In this article, we discuss the alternative design solutions that can be adopted, in presence of late measurements, to support different types of queries that enable meaningful historical analysis. These solutions are based on the enforcement of the distinction between transaction time and valid time within the schema that represents the fact of interest. Besides, we provide a qualitative and quantitative comparison of the solutions proposed, aimed at enabling well-informed design decisions.

INTRODUCTION

Time is commonly understood as a key factor in data warehousing systems since the decisional process often relies on computing historical trends and on comparing snapshots of the enterprise taken at different moments. Within the multidimensional model, time is usually a dimension of analysis; thus, the representation of the history of fact values across a given lapse of time, at a given granularity, is directly supported. For instance, in a relational implementation for the sales domain, for each day there will be a set of rows in the fact table reporting the values of fact QuantitySold on that day for different products and stores. On the other hand, although the multidimensional model does not inherently represent the history of values for dimensions and their properties, some ad hoc techniques were devised to support the so-called *slowly-changing dimensions* (Kimball, 1996). In both cases, time is commonly meant as valid time in the terminology of temporal databases (Jensen et al., 1994) (i.e., it is meant as the time when an event or change *occurred* in the business domain) (Devlin, 1997). Transaction time, meant as the time when the event or change was *registered* in the data warehouse, is typically given little or no importance since it is not considered to be relevant for decision support.

One of the underlying assumptions in data warehouses is that, once an event has been registered (under the form of a row in the fact table), it is never modified so that the only possible writing operation consists in appending new events (rows) as they occur. While this is acceptable for a wide variety of domains, some applications call for a different behavior. In particular, the values measured for a given event may change over a period of time, to be consolidated only *after* the event has been for the first time registered in the data warehouse. This typically happens when the early measurements made for events may be subject to errors (e.g., the amount of an order may be corrected after the order has been registered) or when events inherently evolve over time (e.g., notifications of university enrollments may be received and registered several days after they were issued).

In this context, if the up-to-date situation is to be made timely visible to the decision makers, past events must be continuously updated to reflect the incoming data. Unfortunately, if updates are carried out by physically *overwriting* past registrations of events, some problems may arise:

- Accountability and traceability require the capability of preserving the exact information the analyst based his or her decision upon. If the old registration for an event is replaced by its latest version, past decisions can no longer be justified.
- In some applications, accessing only up-to-date versions of information is not sufficient to ensure the correctness of analysis. A typical case is that of queries requiring to compare the progress of an ongoing phenomenon with past occurrences of the same phenomenon: since the data recorded for the ongoing phenomenon are not consolidated yet, comparing them with past consolidated data may not be meaningful.

Remarkably, the same problems may arise when events are registered in the data warehouse only once, but with a significant delay with respect to the time when they occurred in the application domain (e.g., there may be significant delays in communicating the daily price of listed shares on the stock market): no update is necessary in this case, yet valid time is not sufficient to guarantee accountability. Thus, in more general terms, we will use term *late measurement* to denote any measurement of an event that is sensibly delayed with respect to the time when the event occurs in the application domain; a late measurement may either imply an update to a previous measurement (as in the case of late corrections to orders) or not (as in the case of shares).

In this article, we discuss and compare the design solutions that can be adopted, in presence of late measurements, to enable meaningful historical analysis aimed at preserving accountability and consistency. These solutions are based on the enforcement of the distinction between transaction time and valid time within the schema that represents the fact of interest.

The rest of the article is organized as follows. In the second and third sections, respectively, we survey the related literature and present the working examples. In the fourth section, we distinguish two possible semantics for facts and give definitions of events and registrations. In the fifth section, we distinguish three basic categories of queries from the point of view of their temporal requirements in presence of late measurements. The sixth and seventh sections introduce, respectively, two classes of design solutions: monotemporal and bitemporal, that are then quantitatively compared in the eighth section. The ninth section concludes by discussing the applicability of the solutions proposed.