# A Conceptual Solution for Representing Time in Data Warehouse Dimensions*

**Elzbieta Malinowski**† **Esteban Zimányi**

Department of Computer & Network Engineering
Université Libre de Bruxelles,
50 av. F.D.Roosevelt, 1050 Brussels, Belgium,
E-mail: emalinow,ezimanyi@ulb.ac.be

**Abstract**

Data Warehouses (DWs) use an omnipresent time dimension for keeping track of changes in measure values. However, this dimension cannot be used to model changes in other dimensions. On the other hand, Temporal Databases (TDBs) have been successfully used for modelling time-varying information. Bringing together these two research areas, leading to Temporal Data Warehouses (TDWs), provides the necessary solutions for managing time-varying data in dimensions. In this paper, we introduce temporal extensions for the MultiDimER model, a conceptual multidimensional model. In our model we allow the inclusion of valid and transaction time, which are obtained from source systems, in addition to the data warehouse loading time. Our model allows a conceptual representation of time-varying levels, attributes, and hierarchies. For the latter, we discuss different cases depending on whether the changes in levels affect the relationships between them.

*Keywords:* Data warehouses, conceptual modelling, temporal data warehouse design, time-varying levels, time-varying hierarchies.

## 1 Introduction

Decision-making users increasingly rely on Data Warehouses (DWs) to access historical data for supporting the strategic decisions of organizations. A DW is "a collection of subject-oriented, integrated, non-volatile, and time-variant data to support management's decisions" (Inmon 2002). Subject orientation means that the development of DWs is done according to the analytical necessities of managers on different levels of the decision-making process. Integration represents the complex effort to join data from different operational and external systems. Non-volatility ensures data durability and time-variation indicates the possibility to keep different values of the same information according to its changes in time. Therefore, the last two features indicate that DWs should allow changes to data without overwriting values of already existing data.

The structure of a DW is based on a multidimensional view of data usually represented at a logical level using a *star* or *snowflake* schema, consisting of fact and dimension tables (Figure 1).

A *fact table* (e.g., Sales facts in Figure 1) represents the subject orientation and the focus of analysis, e.g., analysis of sales. It usually contains numeric data called *measures* (e.g., Quantity or Sales in Figure 1) representing analysis needs in quantified form. *Dimensions* (e.g., Product, Time, and Store in Figure 1) are used for exploring the measures from different analysis perspectives. They usually contain hierarchies, such as Product–Category–Department in Figure 1. Further, a dimension may also have descriptive attributes, e.g., Store number or Manager's name in the Store dimension.

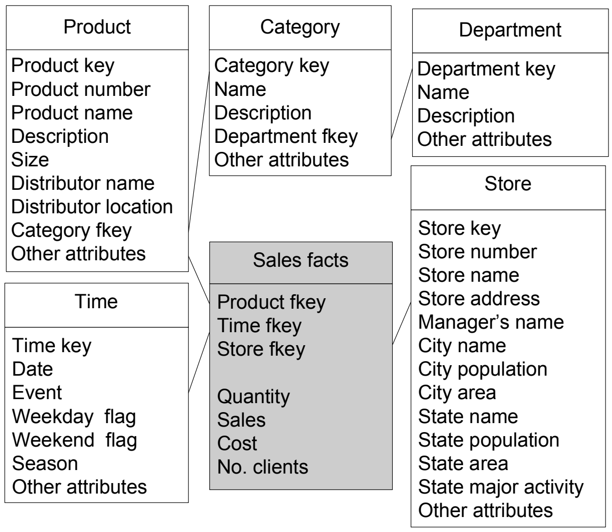


Figure 1: A snowflake schema for a Sales Data Warehouse.

On-Line Analytical Processing (OLAP) systems allow decision-making users to dynamically manipulate the data contained in a DW. OLAP systems use a structure called a *cube* which is also based on dimensions, measures, and hierarchies. Hierarchies allow both detailed and generalized view of data using the roll-up and drill-down operations. Further, the slice and dice operations allow to select a portion of the data based on specified values in one or several dimensions.

Current DW and OLAP models include an omnipresent time dimension that, as the other dimensions, is used for grouping purposes (the roll-up operation) or in a predicate role (the slice and dice operations). For example, if the measure Sales in Figure 1 is represented on a monthly basis, selecting some quarter in the Time dimension will aggregate all monthly measures corresponding to that quarter.

Nevertheless, even though the time dimension additionally serves as a time-varying indicator for measures, e.g., total sales in March 2005, it cannot be used for representing the time when changes

*The work of E. Malinowski was funded by a scholarship of the Cooperation Department of the Université Libre de Bruxelles.

†Currently on leave from the Universidad de Costa Rica.

in other dimensions have occurred (Eder, Koncilia & Morzy 2002). Therefore, usual multidimensional models are not symmetric in the way of representing changes for measures and dimensions: while they allow to track changes in measure values, they are not able to represent changes in dimension data and the time when these changes have occurred, e.g., when a product has changed its ingredients. Consequently, the features of "time-variant" and "non-volatility" only apply for measures leaving to applications the representation of changes occurring in dimensions.

To represent these changes, several implementation solutions were proposed for relational databases for the so-called *slowly-changing dimensions* (Kimball, Ross & Merz 2002). However, some of them do not preserve the entire history of data. Another solution allows to reflect changes to data but requires significant programming effort for managing and querying time-varying dimension data. Further, these solutions do not consider research related to managing time-varying information in temporal databases.

Temporal databases (TDBs) have been investigated over the last decades, e.g., (Elmasri & Wuu 1990, Snodgrass 1995). They provide structures and mechanisms for representing and managing information that vary over time. Two different temporal types[1] are usually considered: *valid time* (VT) and *transaction time* (TT) that allow to represent, respectively, when the data is true in the modelled reality and when it is current in the database. If both temporal types are used, they define *bitemporal time* (BT). Further, in some applications, changes in time can be defined for an object as a whole, i.e., recording *lifespan* (LS) or *existence time* of a database object.

These temporal types are used for representing either *events*, i.e., something that happens at a particular time point, or *states*, i.e., something that has extent over time. For the former an *instant* is used, i.e., a time point on an underlying time axis. A state is represented by an *interval* or *period* indicating the time between two instants using either a non-anchored (e.g., two weeks) or an anchored length of time (e.g., [02/11/2004,05/01/2005]), respectively. Sets of instants and sets of intervals can also be used for representing events and states.

Temporal Data Warehouses (TDWs) join the research achievements of Temporal Databases and Data Warehouses in order to manage time-varying multidimensional data. TDWs raise many issues including consistent aggregation in presence of time-varying data, temporal queries of multidimensional data, storage methods, temporal view materialization, etc. Nevertheless, very little attention from the research community has been drawn to conceptual modelling for TDWs and to the analysis of which temporal support should be included in TDWs considering that TBDs and DWs are semantically different:

- DW data is integrated from existing source systems. TDB data is inserted by users since it represents operational or transactional databases. Therefore, different temporal support may exist according to the types of source systems that integrate data into a TDW.
- DWs support the decision-making process, while TDBs reflect data changes in the reality (VT) and in the database content (TT). To expand the analysis spectrum for decision-making users different temporal types should also be considered in TDWs.

[1]Usually called time dimensions; however, we use the term "dimension" in the multidimensional context.

- DW data is neither modified nor deleted[2]. In contrast, in TDBs users change data directly usually recording the time of these changes as TT. Thus, the TT generated in a TDW plays a different role from the TT used in a TDB.
- DWs are designed according to analysis needs of decision-making users mostly based on a multidimensional view of data with clearly distinguished measures and dimensions. The last two play different roles: measures are aggregated while dimensions are used to explore measures according to different criteria. On the other hand, TDB design is concerned with transactional or operational applications where all data is handled in a similar manner. Therefore, the analysis of temporal support for multidimensional models should consider different aspects present in managing time-varying measures and dimensions.
- Typically DW data reflects measure changes leaving to application programming the representation of changes in dimension data. TDBs allow to express and manage changes for any data. The inclusion of the temporal support for DW data based on TDB research offers the solution for representing and managing in a similar way time-varying dimensions and measures.

Regarding temporal support in TDWs, most works include VT while some of them mention the possibility to have TT or BT support. However, they usually consider TT as the time when the fact is current in a DW, whereas in our model TT as well as VT are incorporated from source systems, if they exist and are required for analysis purposes. Further, in addition to TT and VT, we propose the inclusion of data warehouse loading time (DWLT) indicating since when the data has been current in TDWs.

On the other hand, even though some proposals formally describe the temporal support for a multidimensional model, to our knowledge none of them offer a graphical representation that can be used for communication between users and designers during the design phase of a TDW. Including temporal types in the conceptual model allows to include temporal semantics as an integral part of TDWs.

In this paper we introduce temporal extensions for the MultiDimER model (Malinowski & Zimányi 2005). Due to space limitations, we only refer to levels and hierarchies. Section 2 briefly recalls the main features of the MultiDimER model. Section 3 presents the temporal types included in the model. Section 4 describes our proposal for representing time-varying levels while Section 5 refers to changes occurring in levels as well as in relationships between them. Section 6 introduces a metamodel for a temporal dimension. In Section 7 we present a brief description of the transformation of the temporally extended MultiDimER model into the ER model. Finally, the related works are presented in Section 8 and conclusions are given in Section 9.

## 2 Overview of the MultiDimER model

It has been acknowledged for several decades that conceptual models are the best vehicle for communicating with users during the design process. We proposed the MultiDimER model (Malinowski & Zimányi 2004), a conceptual model based on ER constructs that allows to include different kinds of hierarchies. Figure 2 represents graphical notations used in our model.

[2]We ignore modifications due to errors during data loading and deletion for purging DW data.

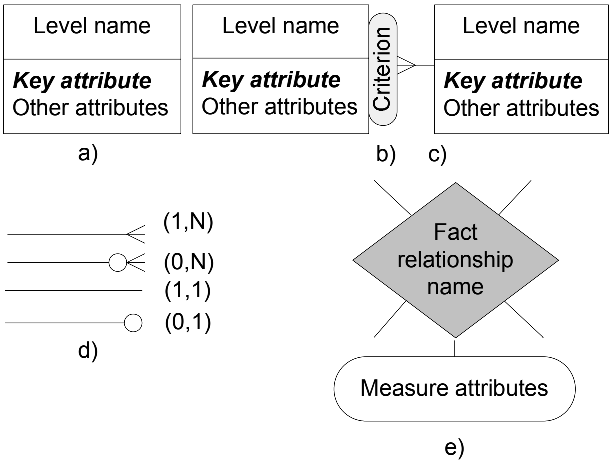


Figure 2: Notations of our multidimensional model: a) one-level dimension, b) analysis criterion, c) hierarchy, d) cardinalities, and e) fact relationship.

We briefly recall the definition of the MultiDimER model[3]. We define a *schema* as a finite set of dimensions and fact relationships. A *dimension* is an abstract concept for grouping data that shares a common semantic meaning within the domain being modelled. It represents either a level, or one or more hierarchies. Levels correspond to entity types (Figure 2 a). Every instance of a level is called a *member*.

*Hierarchies* are required for establishing meaningful paths for roll-up and drill-down operations. They express different structures according to the criteria used for analysis (Figure 2 b), e.g., geographical location or organizational structure. A hierarchy contains several related levels (Figure 2 c). *Cardinalities* (Figure 2 d) indicate the minimum and the maximum numbers of members in one level that can be related to a member in another level. Given two consecutive levels of a hierarchy, the higher level is called *parent* and the lower level is called *child*. A level of a hierarchy that does not have a child level is called *leaf*; the last level, i.e., the one does not have a parent level is called *root*. The root represents the most general view of data.

Levels contain one or several *key attributes* (represented in bold and italic in Figure 2) and may also have other *descriptive attributes*. A key attribute of a parent level defines how child members are grouped. A key attribute in a leaf level or in a level forming a dimension without hierarchy indicates the granularity of measures in the associated fact relationship.

A *fact relationship* (Figure 2 e) represents an *n*-ary relationship between leaf levels. It may contain attributes commonly called *measures*. The latter usually represent numerical data meaningful for leaf members that are aggregated while traversing a hierarchy. Since the roles of a fact relationship always have (0,N) cardinality, we omit such cardinalities to simplify the model.

## 3 Temporal types for TDW

Most works in TDWs, e.g., (Abelló & Martín 2003, Koncilia 2003) include valid time for representing when the data is valid in the modeled reality. This temporal type is important for TDW applications since it allows to aggregate measures correctly as demonstrated in several works, e.g., (Eder et al. 2002).

Further, the usual practice for TDWs is to ignore TT coming from source systems, e.g., (Bliujute, Slatenis, Slivinskas & Jensen 1998, Body, Miquel, Bédard & Tchounikine 2003, Mendelzon & Vaisman 2000). However, in this way traceability applications, e.g., for fraud detection cannot be implemented. Other approaches, e.g., (Abelló & Martín 2003) transform TT from source systems to represent VT in the TDW. This is semantically incorrect because data may be included in databases after their period of validity has expired, e.g., client's previous address.

Moreover, some works, e.g., (Koncilia 2003), consider TT generated in a TDW in the same way as TT is used in TDBs, i.e., it allows to know when data was inserted, modified, or deleted from databases. Nevertheless, TDW data is neither modified nor deleted. Thus, TT in TDWs represents indeed the time when data was loaded into a TDW. This time is called in our model *data warehouse loading time* (DWLT).

DWLT can differ from TT or VT of source systems due to the delay between the time when the changes occurred in source systems and the time when these changes are integrated into a TDW. DWLT is important especially in active DWs (Bruckner & Tjoa 2002) and in creating TDWs from non-temporal sources (Yang & Widom 1998). It also may help to better understand decisions made in the past and adjust loading frequencies if necessary. For example, based on a growing tendency of product sales during weeks 10, 11 and 12 (Figure 3), it was decided to buy more products. However, only in the next DW load, occurred eight weeks later, a new situation has been revealed: a sudden decrease of sales. Thus, an additional analysis can be performed to understand the causes of these changes in sales behaviour. Further, the decision of more frequent loads may be taken.

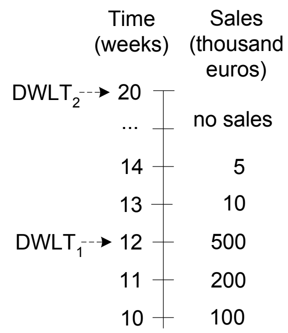


Figure 3: An example of the usefulness of having DWLT.

In some applications not only the validity of member attributes but also the member existence in the modelled reality is important, i.e., its lifespan (LS). The inclusion of LS allows to perform different analysis, e.g., discovering how sales change after the exclusion of some products.

Since current DWs do not offer different temporal types, users may have difficulties in expressing their needs for some kinds of applications, e.g., for fraud detection when TT from a source system is required. Our MultiDimER model meets users' expectations in modelling multidimensional data that vary over time allowing VT, TT, or bitemporal time (BT) coming from source systems and DWLT generated by a TDW. Further, if the source systems include LSs for data representing level members, this temporal type may also be included in a TDW.

In the modelling process, application requirements determine the type of temporal support (none, VT,

[3]The formal semantics of the MultiDimER model based on denotational specification is described in (Malinowski & Zimányi 2005).

TT, BT, DWLT) that needs to be captured in each element of a TDW (attributes, levels, hierarchies, and/or measures). Obviously that depends on whether or not the different data sources of the TDW provide temporal support. These sources may also contain user-defined time attributes playing the role of VT. Therefore, based on the classification of source systems given by Jarke *et al.* (2003) and additionally considering TDBs as another kind of a source system similar to Abelló and Martín (2003), in the following we discuss the temporal support that can be obtained from source systems while building TDWs.

- *Snapshot*: the access to the data in source systems is done through dump of data. To find the changes, the current data is compared with the previous snapshot(s). The time when the snapshot is realized does not determine neither TT nor VT. However, VT may be included as a user-defined attribute.
- *Queryable*: source systems offer a query interface. The detection of changes is done by periodic polling of the data in source systems and by comparing them with the previous version. They may be considered as snapshot systems with the difference of having direct access to data, thus they may contain VT as a user-defined attribute.
- *Logged*: all actions are registered. The periodic polling of data is required for discovering what kinds of changes to which data are applied. The log files contain TT that may be retrieved. Similar to the previous systems, VT may be present.
- *Specific*: each data is a particular case. There is not a general method for data extraction and detection of data changes. These systems can be considered as logged systems if either delta files or timestamps for attributes are available; otherwise, they may be treated as snapshot sources. Therefore, they may include TT and/or VT.
- *Callback or internal actions*: source systems provide triggers, active capabilities, or programming environment so they are able to automatically detect changes of interest and notify those changes to the interested parties, i.e., to a TDW. They offer TT and may include VT.
- *Replicated*: the detection of changes is done by analysing the messages sent by the replication system. This may happen manually, periodically, or using specific criteria. Depending on the features of the change monitor, this kind of systems may offer TT and VT.
- *Bitemporal*: TDBs include the information that allow to know when the objects are valid in reality and when they are current in a DB. Since bitemporal aspect is already represented in source systems, TT and VT are available.

In the following for simplifying the discussion we will refer to VT; the inclusion of TT is straightforward even though it is less used for dimensional data.

## 4 Time-varying levels

Changes in a level can occur either for attribute values (e.g., a product changes its ingredients) or for a member as a whole (e.g., inserting or deleting a product). For the former, we use attribute timestamping since it better represents reality keeping changes only for the specified attributes. For the latter, to indicate the time when a member exists in the modelled reality, i.e., its lifespan, we use the LS symbol next to the level name. A level that includes temporal attributes and/or a lifespan support is called a *temporal level*.

Not all levels or their attributes need to represent changes in time. The MultiDimER model allows to choose which historical data users want to keep by including the symbols of corresponding temporal types for attributes and/or for a level. Notice that the changes to a level member as whole can be represented in the model independently of the fact that the level has temporal attributes.

Figure 4 a) shows an example of an Employee level that includes temporal attributes Position and Title. We group time-varying attributes firstly, to ensure that both kinds of attributes (temporal and non-temporal) can be clearly represented and secondly, to include a smaller number of symbols. For indicating the specific temporal types we use the abbreviations VT, TT, BT, and DWLT.

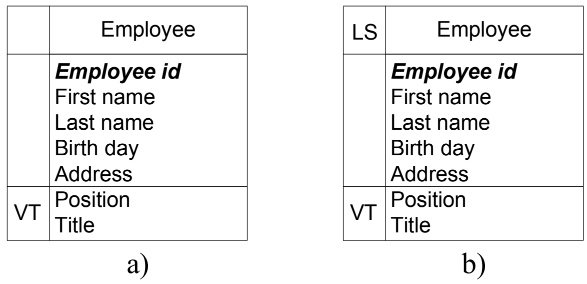


Figure 4: Representation of a) time-varying attributes and b) level lifespan.

The lifespan support for level members (Figure 4 b) indicates that each member includes the LS together with one value per attribute for non-temporal attributes and history of changes for temporal attributes. For example in Figure 4 b) every employee includes the LS together with one value per non-temporal attribute (e.g., Address) and the history of values for Position and Title.

Existing temporal models impose constraints for timestamped attributes and their corresponding object (entity) types, e.g., the VT of attribute values must be within the LS of the object (entity). Our model does not force it. In this way, different situations can be modelled, e.g., a product that does not belong to a store inventory (it is not included in the master file), but it is on sales for defining its acceptance level. For this product, the VT of temporal attributes may not be within the product LS. On the other hand, temporal integrity constraints may be explicitly defined, if required, using a calculus that includes Allen's operators (Allen 1984).

Further, the LS as well as VT used for attributes can be combined with TT or DWLT. In this way, users can obtain the information when the level member or the specific attributes values is current in a source or in a TDW, respectively.

## 5 Time-varying hierarchies

The MultiDimER model allows to represent hierarchies that contain several related levels. Given two consecutive levels in a hierarchy, the levels, the relationship between them or both the levels and the relationship between them may be temporal. We examine next these different situations.

### 5.1 Temporal levels and non-temporal relationships between them

Temporal levels can be associated with non-temporal relationships. Temporality in levels requires to keep changes for attributes or for lifespan of members. On the other hand, non-temporal relationships indicate that either these relationships never change or if they do, only the last modification is kept. An example is given in Figure 5.

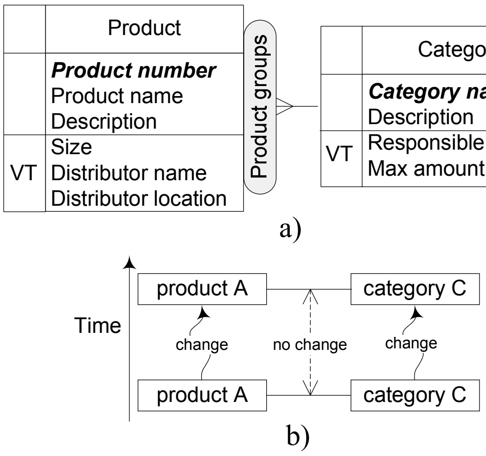


Figure 5: Time-varying levels forming a hierarchy: a) model and b) example of changes.

Nevertheless, if level members change the key attributes used for traversing from one level to another during the roll-up and drill-down operations, incorrect analysis scenario or dangling references may occur. For example, suppose in Figure 5 a) that the Product and Category levels include VTs for key attributes. Product A belongs to category C, but this category is divided in two new categories called C1 and C2, leaving category C invalid from now on. If the relationship product A – category C1 replaces a previous version of product A – category C, the analysis previous to this change will be incorrect, i.e., the measure will be aggregated to a new category C1 that did not exist prior to that change. If the relationship is not modified, references to the invalid version of category C will be made for the sales occurring after the categories have split.

To avoid an incorrect management of hierarchies as described above, we allow temporal levels with non-temporal relationships between them only for those levels that do not keep their LS and/or do not include VT for their key attributes. For example in Figure 5 a) the only changes allowed are those that do not affect relationships between members of these levels, e.g., a product changes its distributor but it belongs to the same category. Figure 5 b) illustrates which changes are allowed.

Notice that DWLT can always be included for levels or attributes since this temporal type only refers to the time when TDW members or attribute values are available for analysis purposes and this time does not affect their validity.

### 5.2 Temporal levels and temporal relationships between them

Temporal levels may include lifespan support and/or time-varying key attributes. This temporal support ensures to keep all changes occurring to level members and/or attribute values. However, these changes may affect the relationships with members of child and/or parent levels. For example, the geographical distribution in Europe during the last 20 years has changed since some countries cease to exist, are merged, or split (Eder et al. 2002). As a consequence, in hierarchies representing this geographical distribution, reassignment of states or provinces to new countries may be required.

Therefore, in the case when levels include LS support and/or VT for key attributes, to avoid incorrect management of hierarchies as described in the previous section, relationships between temporal levels must also be temporal. This restriction in the MultiDimER model as in most TDB models helps to avoid dangling references, i.e., linking to non-existing objects.

The MultiDimER model allows to include VT, TT, BT, and/or DWLT for representing time-varying relationships between levels. We do not include LS for these relationships since these relationships do not exist without their participating levels.

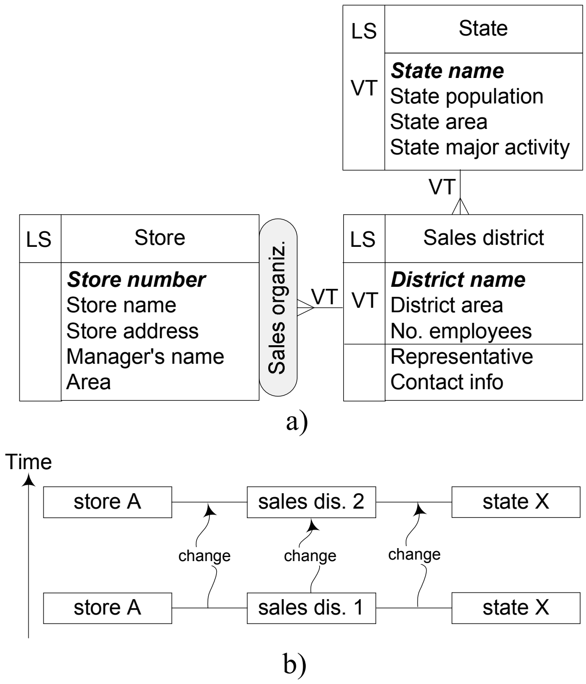


Figure 6: Time-varying levels and time-varying relationship between them: a) model and b) example of changes.

In the example of Figure 6 a) all levels are temporal. They include LS for dimension levels and VT for some attributes. Further, the relationships between levels are also temporal. Suppose that the sales company is in an active development and changes to Sales districts may occur to improve the organizational structure. These changes may affect the relationship with members of the Store and State levels (Figure 6 b).

Further, the constraint for relationships between levels forming a hierarchy is more restrictive than the one usually used for relationships between temporal objects. In TDBs the valid time of a relationship instance must be included in the intersection of the valid times of participating objects. In multidimensional hierarchies it is further required that every valid child (respectively parent) member must be associated with at least one valid parent (respectively child) member in order to ensure correctness of the roll-up and drill-down operations. Validity can refer to the LS of a level as well as to the VT of key attributes leading to the following constraints:

1. Every time point included in the LS of a level must be included in the LS of some member of

the next level, i.e., a valid child member must have a valid parent member and vice versa. If this condition is not fulfilled, structural changes to hierarchies could occur, e.g., forcing some level members to skip the current parent level[4].

2. Every time point included in the VT of a key attribute (i.e., used for aggregation purposes) of a child (respectively parent) member must be included in the VT of some key attribute of a parent (respectively child) member.

### 5.3 Non-temporal levels and temporal relationships between them

Non-temporal levels can be linked with temporal relationships if the values of the level members do not change but the changes to relationship between levels are kept. Some examples of this changing relationships, called *transitions* (Zimányi, Parent & Spaccapietra 1997) include *evolution* and *extension*. The former occurs when a child member ceases to be related to one parent member and is assigned to another one, e.g., a section is assigned to a new division (Figure 7 a). The latter takes place when a child member belongs to the original parent member and additionally a new relationship with a different parent member is included, e.g., the old section is assigned to the new division, leaving it also as a part of the old division (Figure 7 b).

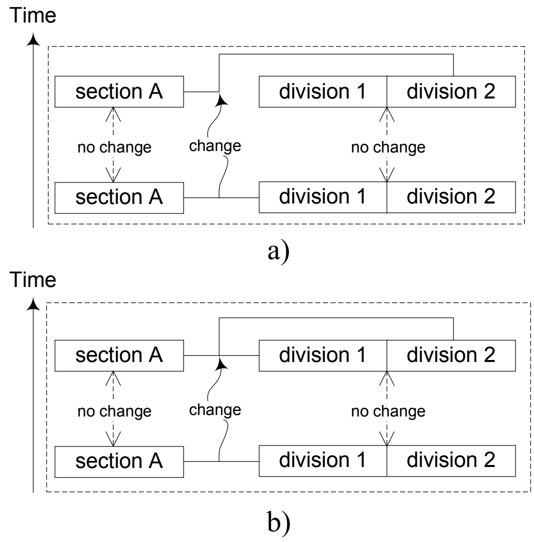


Figure 7: Time-varying relationships between non-temporal levels: a) evolution and b) extension.

To represent temporal support allowing changes in relationships between non-temporal levels we place the corresponding symbol, e.g., VT, on the link between hierarchy levels. To ensure consistency during roll-up and drill-down operations, level members cannot be modified as explained in the previous sections.

### 5.4 Conditions for including a temporal support in hierarchies

Based on the explanations given in the previous sections, in this section we will summarize the conditions for including temporal support in multidimensional hierarchies. Further, we also include cases not seen until now when either temporal or non-temporal relationships exist between a temporal and a non-temporal levels.

1. Temporal levels and non-temporal relationships between them: temporal features can only be applied for attributes that do not participate in the roll-up and drill-down operations.

2. Temporal levels and temporal relationships between them: levels, attributes, and links indicating the relationship between levels can be temporal.

3. Non-temporal levels and temporal relationships between them: no modifications to level members are allowed.

4. Non-temporal levels and non-temporal relationships between them: changes to dimension data cannot be kept. This is the current situation where implementation "tricks" must be used to represent changes to level members and/or to relationships between them.

5. One temporal and one non-temporal level and temporal relationships between them: similar to Case 3, thus non-temporal level members cannot be modified.

6. One temporal and one non-temporal level and non-temporal relationships between them: similar to Case 1, thus temporal level members only can have temporal types for non-key attributes.

### 5.5 Snapshot and lifespan cardinalities

Cardinalities in a non-temporal model indicate the number of members of one level that can be related to member(s) of another level. In our model, this cardinality may be interpreted in two possible ways: the *snapshot cardinality* and the *lifespan cardinality*. The former is considered for every time instant while the latter over its lifespan.

The lifespan cardinality may be different from the snapshot cardinality because of the changes in hierarchies, both in levels and in relationships between them. Thus, when these temporal changes must be kept, both lifespan and snapshot cardinalities may be considered.

In the MultiDimER model the snapshot cardinality is by default equal to the lifespan cardinality; however, if these cardinalities are different, a dotted line with the LC symbol is inserted and it indicates the lifespan cardinality as shown in Figure 8.

Further, the constraint imposed on the cardinalities requires the minimum value as well as the maximum value of the lifespan cardinalities to be equal or greater than minimum and maximum values of the snapshot cardinalities, respectively.

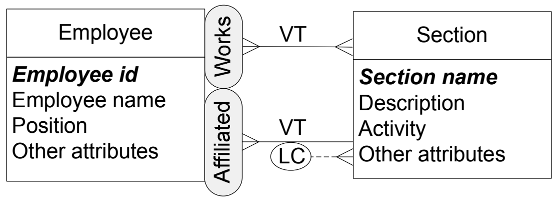


Figure 8: Snapshot and lifespan cardinalities between hierarchy levels.

In the example in Figure 8, the employee snapshot and lifespan cardinalities for the hierarchy Works are many-to-many indicating that an employee can work in more than one section at the same time instant

[4] We do not consider structural changes to hierarchies since they require schema versioning that is out of the scope of this paper.

and over his lifespan. On the other hand, the snapshot cardinality for the hierarchy Affiliated is one-to-many, and the lifespan cardinality is many-to-many indicating that in every time instant an employee can be affiliated only to one section, but over his lifespan he can be affiliated to many sections.

## 6 Metamodel of a temporally-extended dimension in the MultiDimER model

We give next a metamodel for a dimension of the temporally-extended MultiDimER model - a conceptual model used to represent dimensions, hierarchies, and levels with attributes, which may change over time.

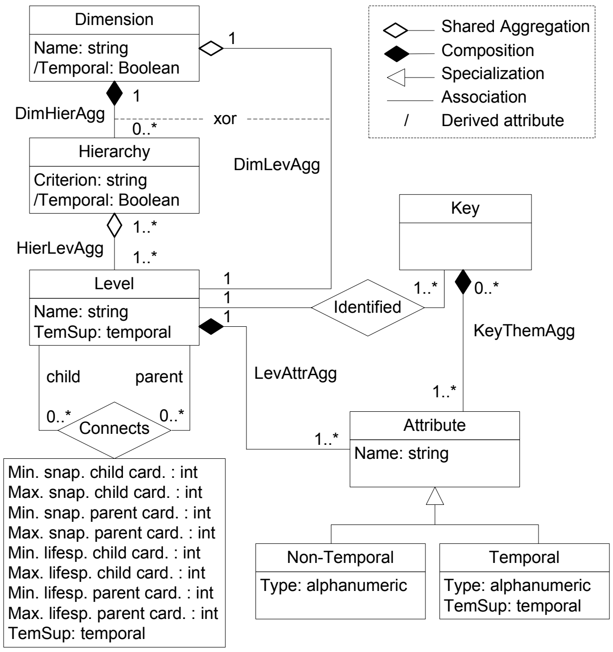


Figure 9: Metamodel of a dimension.

As shown in Figure 9, a dimension is comprised of either a level, or one or more hierarchies. A hierarchy contains several related levels. These levels are associated through child-parent relationship. Levels include attributes, some of which are key attributes used for aggregation purposes.

We define a *temporal level* as a level for which the application needs to keep its time-varying characteristics. This is captured by including different temporal types for attributes and/or for a level, i.e., the TemSup attribute in Figure 9. We allow VT, TT, or BT coming from source systems (if available) and DWLT generated by DBMS of a TDW. Notice, that VT for a level is represented by its LS.

Further, the relationship between levels may also be temporal independently of whether the levels are temporal or not. This is indicated in Figure 9 by the attribute called TempSup for the Connects relationship between child and parent levels. The model allows to include VT, TT, BT, and/or DWLT for this relationship.

Additionally, the relationship between two levels is characterized by cardinalities, which indicate the minimum and the maximum number of members in one level that can be related to a member in another level. We distinguish the snapshot cardinality and lifespan cardinality. The former is the cardinality in a instant of time whereas the latter represents this cardinality over members lifespan.

A dimension is temporal if it has at least one temporal hierarchy. A hierarchy is temporal if it has at least one temporal level or one temporal relationship between levels. Since temporal hierarchies (respectively dimensions) can combine temporal and non-temporal levels (respectively hierarchies), we call a hierarchy (respectively dimension) *fully temporal* when all its levels and relationships between them (respectively hierarchies) are temporal. It is called *partly temporal* when it contains at least one non-temporal level or one non-temporal relationship between levels (respectively one non-temporal hierarchy).

## 7 Transformations to the ER model

The MultiDimER model can be implemented by mapping its specifications into those of operational data models. We already proposed the mapping into relational (Malinowski & Zimányi 2005) and object-relational (Malinowski & Zimányi 2006) data models. Further, another two-phase approach may be adopted where a MultiDimER schema is transformed into a conventional ER schema and afterwards, into a logical schema.

In this section, we briefly describe and give examples of mapping the temporally-extended MultiDimER model into the ER model. The ER model is a widely-used and platform-independent conceptual model. In addition, well-known rules for transformation of ER model into relational model exist, e.g., (Elmasri & Navathe 2003).

The MultiDimER model is mainly based on the ER constructs with their usual semantics, i.e., entity types, attributes, relationship types. Some additional semantics is provided for different kinds of hierarchies (Malinowski & Zimányi 2004) that is beyond the scope of this paper.

In the MultiDimER model a level corresponds to an entity type in the ER model. Non-temporal attributes are represented as monovalued attributes. The temporal support in the MultiDimER model is added in an implicit manner (Gregersen & Jensen 1998), i.e., the timestamp attributes used for capturing a temporal aspect are hidden using instead pictograms. Therefore, the transformation of the time-related data into classical non-temporal structures of the ER model requires additional attributes for timestamps.

The different temporal types can be represented in the ER model as follows: (1) a simple composite attribute for a period (Figure 10 a), (2) a multivalued composite attribute for a set of periods (Figure 10 b), (2) a monovalued attribute for an instant (Figure 10 c), and (3) a multivalued attribute for a set of instants (Figure 10 d). Notice that a set of periods or instants are used when the attribute has the same value in different periods or instants of time.

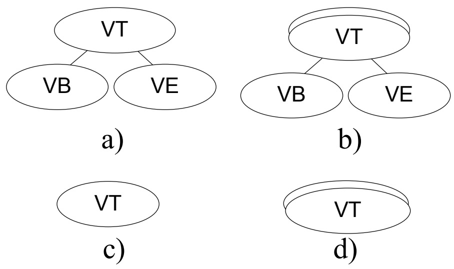


Figure 10: Different representations of VT in the ER model.

Figure 11 shows an Employee level and its corresponding ER diagram. The mapping of each temporal attribute requires a multivalued composite attribute;

it includes an attribute for which the temporal type is attached (Name for Position and Title attributes in Figure 11 b) and an additional attribute for a temporal type. The latter is represented in Figure 11 b) using a set of periods indicating that an employee can have the same position or title in different periods of time.

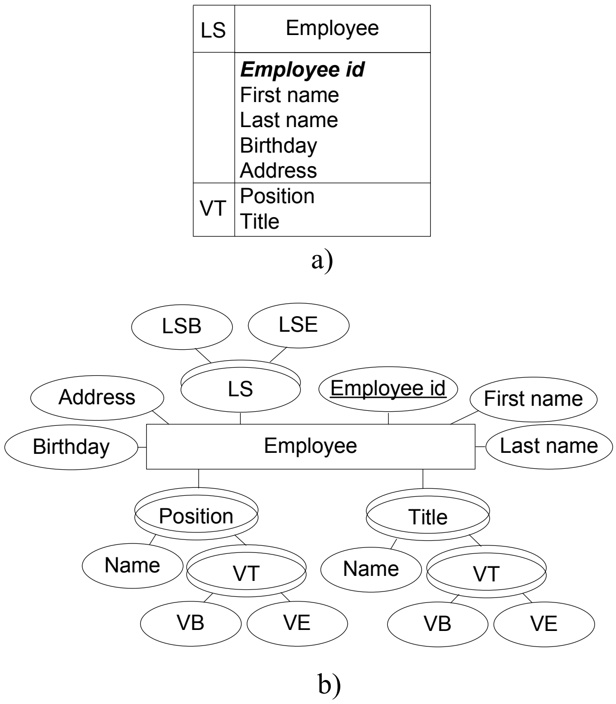


Figure 11: A level with temporal attributes: a) the MultiDimER model and b) corresponding ER model.

Further, as explained before, the MultiDimER model can represent temporal changes to a level member as whole. This is expressed using the LS symbol next to the level name (Figure 11 a). This temporal support is mapped into the ER model using an additional multivalued composite attribute containing the begin (LSB) and the ending (LSE) instants of the lifespan (Figure 11 b). We used a temporal element, i.e., a set of periods since this allows to represent discontinuous lifespan, e.g., a professor leaving for sabbatical during some period of time.

Relationships linking the levels of a hierarchy in the MultiDimER model are usual binary relationships in the ER model. Since this relationship in our model can be temporal, the corresponding binary relationship in the ER model should include an attribute (or several depending on the applied temporal support) in a similar way as was explained for time-varying attributes of a level. For example, the mapping of a temporal relationship between Store and Sales district levels for the example in Figure 6 is shown in Figure 12[5].

In this case, the snapshot and lifespan cardinalities are the same, i.e., many-to-one. If these cardinalities are different, the highest cardinalities is mapped to the ER model; according to the previously specified constraint in Section 5.5, the lifespan cardinality will be mapped.

As can be seen in Figure 11 and 12, the MultiDimER model provides better conceptual representation of time-varying attributes, levels, and relationships. It contains less elements, it clearly allows to distinguish which data changes should be kept, and it leaves outside of user's concerns some more technical aspects such as multivalued or composite attributes.

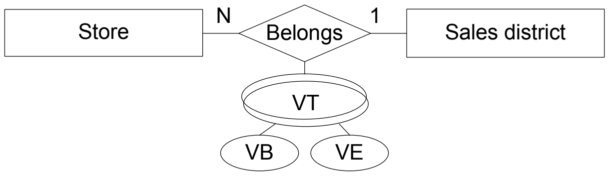


Figure 12: ER representation of temporal relationships between levels in the MultiDimER model.

## 8 Related work

The necessity to manage time-varying data has been acknowledged for several decades, e.g., (Snodgrass 1995). However, no such consensus has been reached for representing time-varying multidimensional data considering the particularities of DW semantics.

Works related to TDWs raise many issues, e.g., the inclusion of temporal types in TDWs, e.g., (Abelló & Martín 2003, Bruckner & Tjoa 2002), temporal querying of multidimensional data, e.g., (Mendelzon & Vaisman 2003, Pedersen, Jensen & Dyreson. 2001), correct aggregation in presence of data and structural changes, e.g., (Eder et al. 2002, Hurtado, Mendelzon & Vaisman 1999, Mendelzon & Vaisman 2003), temporal view materialization from non-temporal sources, e.g., (Yang & Widom 1998), evolution of a multidimensional structure, e.g., (Body et al. 2003, Eder et al. 2002, Mendelzon & Vaisman 2003), or implementation considerations for a temporal star schema, e.g., (Bliujute et al. 1998).

In the following we refer to works that (1) propose different temporal types for TDWs and (2) offer conceptual models for TDWs.

The inclusion of different temporal types in TDWs is briefly mentioned in several works. Most of them consider VT (Body et al. 2003, Bliujute et al. 1998, Eder et al. 2002, Mendelzon & Vaisman 2003, Ravat & Teste 2000, Yang & Widom 1998); some of them mention that it is easy to incorporate TT without giving a deeper analysis (Mendelzon & Vaisman 2003, Pedersen et al. 2001). Other authors consider TT generating it either in a TDW (Abelló & Martín 2003, Koncilia 2003) or of in a given source system (Bruckner & Tjoa 2002)[6].

A more extensive analysis of temporal types for TDWs is given by Abelló and Marín (2003). They discuss the inclusion of TT and VT in TDWs taking into account different types of sources that integrate data in a TDW. VT is calculated based on TT of either a DW or a source. The exception is made for sources based on TDBs, considering only one temporal type, e.g., VT and converting another one, e.g., TT into a user-defined attribute. However, Abelló and Marín (2003) do not consider possible existence of user-defined time attributes in source systems, which may serve for establishing VT in TDWs. Also, TT from source systems is ignored or transformed for representing VT in TDWs. We do not consider TT as a possible approximation of VT, since data can be included and be current in DBs after its validity has expired, e.g., courses taught five years ago.

Therefore, even though most works include VT and some mention the possibility to have TT or explicitly present BT support, they usually considered TT as time where a fact is current in DW, whereas in our model TT as well as VT are incorporated from

[5] For simplicity we do not present level attributes in the figure.

[6] It is called *revelation time* in (Bruckner & Tjoa 2002).

source systems. Further, only Bruckner and Tjoa (2002) discuss the inclusion of VT, TT, and DWLT for active data warehouses, however, they do not offer a conceptual model for a TDW that includes these temporal types.

Several works are dedicated to conceptual modelling of TDWs. Body *et al.* (2003) define a conceptual TDW model that allows a member to have several valid member versions for a given time (when VT overlaps). Further, they include a temporal relationship that establishes an explicit link between two member versions and represents the roll-up function. Since a dimension is a set of member versions and a set of temporal relationships between these members, a temporal dimension is considered as a directed graph where nodes are member versions and arcs are relationships.

Eder *et al.* (2002) propose a temporal multidimensional model called COMET; it allows to represent changes at the schema and instance levels. The model includes VT for members and for relationships between members forming hierarchies. They include a list of constraints to ensure the integrity of their model. Further, in order to reduce incorrect OLAP results due to the dimension changes, their model includes transformation functions given by the user. The COMET model was extended by Koncilia (2003) including TT of a TDW.

Mendelzon and Vaisman (2003) propose a temporal multidimensional model that reuses results from the TDB community. They formally define temporal dimension schema and instances as well as a temporal fact table, which are used for defining temporal multidimensional database. Using VT they build a TOLAP query language that allows the user to choose the way data should be aggregated.

Further, Pedersen *et al.* (2001) extend the basic multidimensional model by temporal support. Their model allows the inclusion of VT as well as TT. These temporal types can be used to express changes in dimension members including their representation, in hierarchy links, and in fact-dimension relationships.

On the other hand, Ravat and Teste (2000) define a DW model using object-oriented approach; it allows to integrate temporal and archive data. Temporal data are used for storing the detailed data evolution while archive data store the summarized data evolutions.

In general, these models formally describe the temporal support for multidimensional models, allowing to express changes in dimension members, hierarchy links, and in fact relationships. However, none of them offer a graphical representation based on a multidimensional view of temporal data that can be used for communication between users and designers. Further, they do not consider different aspects as proposed in this work, e.g., a hierarchy that may have temporal and non-temporal levels linked with either temporal or non-temporal relationships.

## 9 Conclusions

Bringing together two research areas, Data Warehouses (DWs) and Temporal Databases (TDBs), allows to combine the achievements of each of them leading to the emerging field of Temporal Data Warehouses (TDWs). Nevertheless, neither DWs nor TDBs have a well-accepted conceptual model that can be used for capturing users' requirements.

In this paper, we proposed a temporal extension of the MultiDimER model for representing time-varying levels and hierarchies. This model allows to represent both temporal and time-invariant levels and hierarchies. In this way, users and designers are able to choose and express in a unambiguous way which elements they want to be time invariant and for which data they want to express changes occurred in time, as recommended by Gregersen and Jensen (1998).

We included in the model valid and transaction time coming from source systems and the data warehouse loading time generated by a TDW. In this way, users can traverse hierarchies knowing when data is valid in the modelled reality, expand the analysis to traceability applications, and know since when data has been available in a TDW. Further, having lifespan for level members allows to know how the exclusion or inclusion of different members may affect measure values.

However, the inclusion of different temporal types depends on users' requirements and also on their availability in source systems. Taking into account different kinds of source systems, we discussed which temporal support they can offer.

Next, we proposed to extend the MultiDimER model adding time-varying attributes and lifespan of a level. We also discussed three different cases for time-varying hierarchies: (1) temporal levels with non-temporal relationships between them, (2) temporal relationships between temporal levels, and (3) temporal relationships between non-temporal levels. In the first case, we did not allow the modification for key attributes participating in the roll-up and drill-down operations. For the second case we established constraints ensuring that every valid child (respectively parent) member is related to its valid parent (respectively child) member. In this way, changes to members as well as the relationship between them are considered avoiding dangling references. For the third case we imposed the restriction that level members cannot be modified. Further, we included in the model the snapshot and lifespan cardinalities indicating the number of members of one level that can be related to members of another level in every time instant and over its lifespan, respectively.

Finally, we presented the metamodel of dimensions where levels as well as relationships between them may vary over time. We finished describing the transformation of the constructs of the MultiDimER model into the ER model.

Proposing the inclusion of temporal types in a conceptual model allows to include temporal semantics as an integral part of TDWs. Afterwards, logical and physical models can be derived from such a conceptual representation. Further, these logical models can be obtained using a direct mapping of the MultiDimER constructs (Malinowski & Zimányi 2005, Malinowski & Zimányi 2006) or using a two-phase approach: first to the well-known ER model and then to a chosen logical representation.

This work belongs to a larger project aiming at developing a methodology for conceptual design of spatio-temporal data warehouses.

## References

Abelló, A. & Martín, C. (2003), A bitemporal storage structure for a corporate data warehouse, *in* 'Proc. of the 5th Int. Conf. on Enterprise Information Systems', pp. 177–183.

Allen, J. (1984), 'Towards a general theory of action and time', *Artificial Intelligence* **23**(2), 123–154.

Bliujute, R., Slatenis, S., Slivinskas, G. & Jensen, C. (1998), Systematic change mangement in dimensional data warehousing, Technical report, Time Center, TR-23.

Body, M., Miquel, M., Bédard, Y. & Tchounikine, A. (2003), Handling evolution in multidimensional structures, *in* 'Proc. of the 19th Int. Conf. on Data Engineering', pp. 581–592.

Bruckner, R. & Tjoa, A. (2002), 'Capturing delays and valid times in data warehouses – towards timely consistent analyses', *Journal of Intelligent Information Systems* **19**(2), 169–190.

Eder, J., Koncilia, C. & Morzy, T. (2002), The COMET metamodel for temporal data warehouses, *in* 'Proc. of the 14th Int. Conf. on Advanced Information Systems Engineering', pp. 83–99.

Elmasri, R. & Navathe, S. (2003), *Fundamentals of Database Systems*, fourth edn, Adison-Wesley.

Elmasri, R. & Wuu, G. (1990), A temporal model and query language for ER databases, *in* 'Proc. of the 6th Int. Conf. on Data Engineering', pp. 76–83.

Gregersen, H. & Jensen, C. (1998), Conceptual modeling of time-varying information, Technical report, Time Center, TR-35.

Hurtado, C., Mendelzon, A. & Vaisman, A. (1999), Maintaining data cubes under dimension updates, *in* 'Proc. of the 15th Int. Conf. on Data Engineering', pp. 346–355.

Inmon, W. (2002), *Building the Data Warehouse*, John Wiley & Sons.

Jarke, M., Lenzerini, M., Y.Vassiluiou & Vassiliadis, P., eds (2003), *Fundamentals of Data Warehouse*, Springer.

Kimball, R., Ross, M. & Merz, R. (2002), *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, John Wiley & Sons.

Koncilia, C. (2003), A bi-temporal data warehouse model, *in* 'Proc. of Short Papers of the 15th Int. Conf. on Advanced Information Systems Engineering', pp. 77–80.

Malinowski, E. & Zimányi, E. (2004), OLAP hierarchies: A conceptual perspective, *in* 'Proc. of the 16th Int. Conf. on Advanced Information Systems Engineering', pp. 477–491.

Malinowski, E. & Zimányi, E. (2005), Hierarchies in a multidimensional model: from conceptual modeling to logical representation. Accepted for publication in Data & Knowledge Engineering.

Malinowski, E. & Zimányi, E. (2006), Object-relational representation of a conceptual model for temporal data warehouses. Submitted to publication.

Martín, C. & Abelló, A. (2003), A temporal study of data sources to load a corporate data warehouse, *in* 'Proc. of the 5th Int. Conf. on Data Warehousing and Knowledge Discovery', pp. 109–118.

Mendelzon, A. & Vaisman, A. (2000), Temporal queries in OLAP, *in* 'Proc. of the 26th Very Large Database Conference', pp. 243–253.

Mendelzon, A. & Vaisman, A. (2003), Time in multidimensional databases, *in* M. Rafanelli, ed., 'Multidimensional Databases: Problems and Solutions', Idea Group Publishing, pp. 166–199.

Pedersen, T., Jensen, C. & Dyreson., C. (2001), 'A foundation for capturing and querying complex multidimensional data', *Information Systems* **26**(5), 383–423.

Ravat, F. & Teste, O. (2000), A temporal object-oriented data warehouse model, *in* 'Proc. of the 11th Int. Conf. on Database and Expert Systems', pp. 583–592.

Snodgrass, R., ed. (1995), *The TSQL2 Temporal Query Language*, Kluwer Academic Publishers.

Yang, J. & Widom, J. (1998), Mantaining temporal views over non-temporal information source for data warehousing, *in* 'Proc. of the 6th Int. Conf. on Extending Database Technology', pp. 389–403.

Zimányi, E., Parent, C. & Spaccapietra, S. (1997), TERC+: a temporal conceptual model, *in* 'Proc. of the Int. Symp. on Digital Media Information'.