# A Conceptual Model of Temporal Data Warehouses and its Transformation to the ER and Object-Relational Models [⋆]

E. Malinowski [a,1,*] E. Zimányi [a]

[a] *Department of Computer & Network Engineering*
*Université Libre de Bruxelles*
*1050, Brussels, Belgium*

## 1 Formalization of the MultiDim model

The following formalization is inspired from [2]. We first describe notations, assumptions, and meta-variables required for defining the abstract syntax and the semantics of the MultiDim model. Next, we give the abstract syntax of the model that allows the translation from the graphical representation to the equivalent textual representation. Finally, after describing the auxiliary functions, we define the semantics of the MultiDim model.

### *1.1 Notations*

We use *SET*, *FSET*, and *TF* to denote the class of sets, the class of finite sets, and the class of total functions, respectively. Given $S_1, S_2, \ldots, S_n \in SET$, $S_i \uplus S_j$ indicates the disjoint union of sets, $S_i \cup S_j$ denotes the union of sets, and $S_1 \times S_2 \times \ldots \times S_n$ represents the Cartesian product over the sets $S_1, S_2, \ldots, S_n$. $\mathcal{P}(S)$ indicates powerset of the set $S$.

We write finite sets as $\{c_1, c_2, \ldots, c_n\}$, lists as $\langle c_1, c_2, \ldots, c_n \rangle$, and elements

⋆ The work of E. Malinowski was funded by the Cooperation Department of the Université Libre de Bruxelles.
* Corresponding author.
*Email addresses:* emalinow@ulb.ac.be (E. Malinowski), ezimanyi@ulb.ac.be (E. Zimányi).
1 Currently on leave from the Universidad de Costa Rica.

of Cartesian product as $(c_1, c_2, \ldots, c_n)$. For any set, we use $\perp$ to denote the undefined value of the set.

### 1.2 Predefined data types

A data signature describes the predefined data types, operations, and predicates. The MultiDim model includes the basic data types *int*, *real*, and *string*; the inclusion of other data types is straightforward. These predefined data types as well as the operations and predicates on these have the usual semantics, and this interpretation is fixed, that is, defined once and for all.

The syntax of a data signature *DS* be given as follows:

- the sets $\mathit{DATA}, \mathit{OPNS}, \mathit{PRED} \in \mathit{FSET}$,
- a function $\mathit{input} \in \mathit{TF}$ such that $\mathit{input} : \mathit{OPNS} \rightarrow \mathit{DATA}^*$,
- a function $\mathit{output} \in \mathit{TF}$ such that $\mathit{output} : \mathit{OPNS} \rightarrow \mathit{DATA}$, and
- a function $\mathit{args} \in \mathit{TF}$ such that $\mathit{args} : \mathit{PRED} \rightarrow \mathit{DATA}^+$.

If $\sigma \in \mathit{OPNS}$, $\mathit{input}(\sigma) = \langle d_1, \ldots, d_n \rangle$ and $\mathit{output}(\sigma) = d$, this is denoted as $\sigma : \langle d_1, \ldots, d_n \rangle \rightarrow d$. If $\pi \in \mathit{PRED}$, with $\mathit{args}(\pi) = \langle d_1, \ldots, d_n \rangle$, this is denoted as $\pi : \langle d_1, \ldots, d_n \rangle$.

The predefined data types and some operators and predicates on them are as follows.

$$
\begin{aligned}
\mathit{DATA} &\supseteq \{\mathit{int}, \mathit{real}, \mathit{string}\} \\
\mathit{OPNS} &\supseteq \{\, \begin{array}[t]{lll}
+_i, -_i, *_i : & \mathit{int} \times \mathit{int} & \rightarrow \mathit{int} \\
+_r, -_r, *_r : & \mathit{real} \times \mathit{real} & \rightarrow \mathit{real} \\
/_i : & \mathit{int} \times \mathit{int} & \rightarrow \mathit{real} \\
/_r : & \mathit{real} \times \mathit{real} & \rightarrow \mathit{real} \\
\mathit{cat} : & \mathit{string} \times \mathit{string} & \rightarrow \mathit{string} \\
\ldots\} & &
\end{array} \\
\mathit{PRED} &\supseteq \{\, \begin{array}[t]{ll}
<_i, >_i, \leq_i, \geq_i, \neq_i: & \mathit{int} \times \mathit{int} \\
<_r, >_r, \leq_r, \geq_r, \neq_r: & \mathit{real} \times \mathit{real} \\
<_s, >_s, \leq_s, \geq_s, \neq_s: & \mathit{string} \times \mathit{string} \\
\ldots\} &
\end{array}
\end{aligned}
$$

### *1.3 Meta variables*

$S_D \in$ *Schema_DECL* – MultiDim schema declarations
$D_D \in$ *Dim_DECL* – dimension declarations
$L_D \in$ *Lev_DECL* – level declarations
$CP_D \in$ *CPRel_DECL* – child-parent relationship declarations
$F_D \in$ *FactRel_DECL* – fact relationship declarations
$IC_D \in$ *IC_DECL* – integrity constraints declarations
$H_D \in$ *Hier_DECL* – hierarchy declarations
$A_D \in$ *Att_DECL* – attribute declarations
$CP_S \in$ *CP_SPEC* – the set of child-parent specifications
$I_S \in$ *Inv_SPEC* – the set of level involvement specifications
$T_S \in$ *Temp_SPEC* – the set of specifications for temporal support
$D \in$ *Dimensions* – the set of dimension names
$F \in$ *FactRels* – the set of fact relationship names
$L \in$ *Levels* – the set of level names
$CP \in$ *CPRels* – the set of child-parent relationship names
$H \in$ *Hier* – the set of hierarchy names
$A \in$ *Attributes* – the set of attribute names
$K \in 2^{Attributes}$ – the set of subsets of attribute names
$d \in$ *DATA* – the set of basic data types supported by the MultiDim model
$min, max \in$ *Integer constants* – the set of integer constants
$temp \in \{LS, VT, TT, LT\}$ – the set of temporality types
$t \in \{$*Time*,*SimpleTime*,*ComplexTime*,*Instant*,*InstantSet*,*Interval*,
*IntervalSet*$\}$ – the set of temporal data types
$gr \in \{sec, min, hour, day, week, month, year\}$ – the set of granules for temporality

### *1.4 Abstract syntax*

$$
\begin{array}{rcl}
S_D & ::= & D_D; L_D; CP_D; F_D; IC_D; \\
D_D & ::= & D_{D_1}; D_{D_2} \\
 & & \mid \textbf{Dimension } D \textbf{ includes level } L \\
 & & \mid \textbf{Dimension } D \textbf{ includes } H_D \\
L_D & ::= & L_{D_1}; L_{D_2} \\
 & & \mid \textbf{Level } L \textbf{ has } A_D \\
 & & \mid \textbf{Level } L \textbf{ with temporality } T_S \textbf{ has } A_D \\
CP_D & ::= & CP_{D_1}; CP_{D_2} \\
 & & \mid \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
 & & \mid \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
 & & \quad\quad \textbf{has distributing factor}
\end{array}
$$

$$
\begin{array}{rcl}
 & & \mid \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
 & & \qquad \textbf{with temporality } T_S \\
 & & \mid \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
 & & \qquad \textbf{with temporality } T_S \textbf{ has distributing factor} \\
F_D & ::= & F_{D_1}; F_{D_2} \\
 & & \mid \textbf{Fact relationship } F \textbf{ involves } I_S \\
 & & \mid \textbf{Fact relationship } F \textbf{ involves } I_S \textbf{ has } A_D \\
IC_D & ::= & IC_{D_1}; IC_{D_2} \\
 & & \mid K \textbf{ is primary key of } L \\
 & & \mid \textbf{Participation of } L \textbf{ in } CP \textbf{ is } (min, max) \\
 & & \mid \textbf{Snapshot participation of } L \textbf{ in } CP \textbf{ is } (min, max) \\
 & & \mid \textbf{Lifespan participation of } L \textbf{ in } CP \textbf{ is } (min, max) \\
 & & \mid \textbf{Exclusive participation of } L \textbf{ in } CP_S \\
H_D & ::= & H_{D_1}, H_{D_2} \\
 & & \mid \textbf{hierarchy } H \textbf{ composed of } CP_S \\
A_D & ::= & A_{D_1}, A_{D_2} \\
 & & \mid \textbf{Attribute } A \textbf{ of } A'_D \\
A'_D & ::= & \textbf{type } d \\
 & & \mid \textbf{ type } d \textbf{ with temporality } T_S \\
CP_S & ::= & CP_{S_1}, CP_{S_2} \\
 & & \mid CP \\
I_S & ::= & I_{S_1}, I_{S_2} \\
 & & \mid L \\
 & & \mid L \textbf{ as } role \\
T_S & ::= & T_{S_1}, T_{S_2} \\
 & & \mid (temp, t, gr) \\
d & ::= & int \mid real \mid string \\
temp & ::= & LS \mid VT \mid TT \mid LT \\
t & ::= & Time \mid SimpleTime \mid ComplexTime \mid Instant \mid Interval \\
 & & \mid InstantSet \mid IntervalSet \\
gr & ::= & sec \mid min \mid hour \mid day \mid month \mid year
\end{array}
$$

### *1.5 Examples using the abstract syntax*

In this section we show examples of the textual representation of a schema for temporal data warehouses. For brevity, only part of the textual representation is given.

The textual representation of the schema in Figure 1 is given next.

#### *1.5.1 Level definitions*

**Level** *Branch* **has**

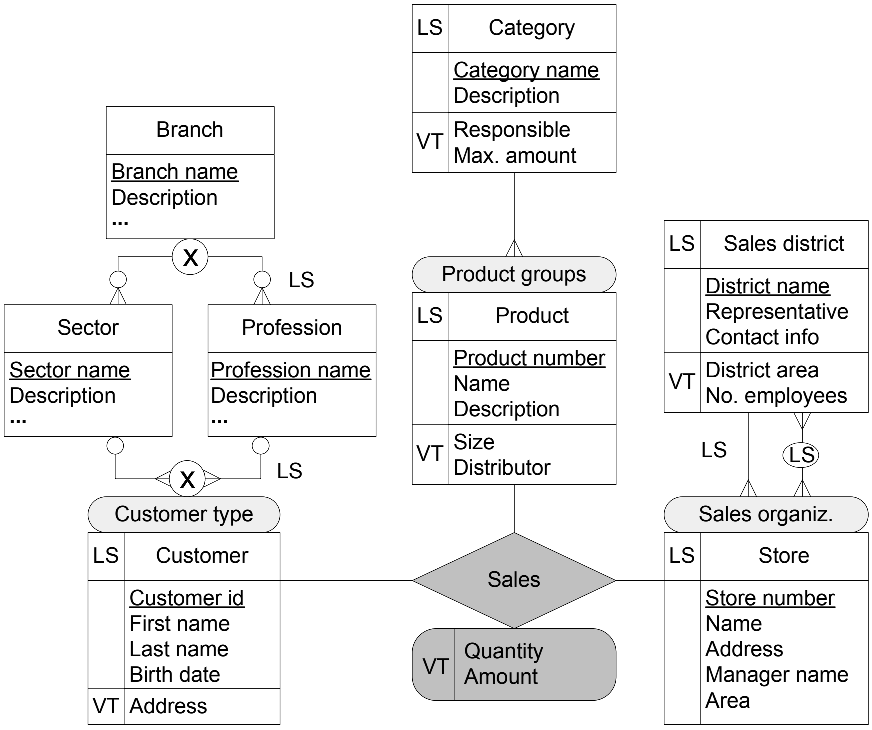


Fig. 1. Example of a schema including temporal and non-temporal elements.

> **Attribute** *Branch name* **of type** *string*,
> **Attribute** *Description* **of type** *string*,
> . . .;

**Level** *Sector* **has**

> **Attribute** *Sector name* **of type** *string*,
> **Attribute** *Description* **of type** *string*,
> . . .;

**Level** *Profession* **has**

> **Attribute** *Profession name* **of type** *string*,
> **Attribute** *Description* **of type** *string*,
> . . .;

**Level** *Customer* **with temporality** $(LS, IntervalSet, month)$ **has**

> **Attribute** *Customer id* **of type** *integer*,
> **Attribute** *First name* **of type** *string*,
> **Attribute** *Last name* **of type** *string*,
> **Attribute** *Birth date* **of type** *string*,
> **Attribute** *Address* **of type** *string*
> **with temporality** $(VT, IntervalSet, month)$,

**Level** *Product* **with temporality** $(LS, IntervalSet, month)$ **has**

> **Attribute** *Product number* **of type** *integer*,
> **Attribute** *Name* **of type** *string*,
> **Attribute** *Description* **of type** *string*,
> **Attribute** *Size* **of type** *real*

**with temporality** $(VT, IntervalSet, month)$,
**Attribute** *Distributor* **of type** *string*
**with temporality** $(VT, IntervalSet, month)$;
**Level** *Store* **with temporality** $(LS, IntervalSet, year)$ **has**
**Attribute** *Store number* **of type** *integer*,
**Attribute** *Name* **of type** *string*,
**Attribute** *Address* **of type** *string*,
**Attribute** *Manager name* **of type** *string*,
**Attribute** *Area* **of type** *string*;
. . .;

#### *1.5.2 Child-parent relationship definitions*

**C-P relationship** *SectBra* **involves** *Sector*, *Branch*;
**C-P relationship** *ProfBra* **involves** *Profession*, *Branch*
**with temporality** $(LS, IntervalSet, month)$;
**C-P relationship** *CustSect* **involves** *Customer*, *Sector*;
**C-P relationship** *CustProf* **involves** *Customer*, *Profession*
**with temporality** $(LS, IntervalSet, month)$;
**C-P relationship** *ProdCat* **involves** *Product*, *Category*;
**C-P relationship** *StoSD* **involves** *Store*, *Sales district*
**with temporality** $(LS, IntervalSet, month)$;

#### *1.5.3 Dimension definitions*

**Dimension** *Product* **includes**
**hierarchy** *Product groups* **composed of** *ProdCat*;
**Dimension** *Customer* **includes**
**hierarchy** *Customer type* **composed of** *CustSect*, *CustProf*,
*SectBra*, *ProfBra*;
**Dimension** *Store* **includes**
**hierarchy** *Sales organiz.* **composed of** *StoSD*;

#### *1.5.4 Fact relationship definitions*

**Fact relationship** *Sales* **involves**
*Customer*, *Product*, *Store*;
**has**
**Attribute** *Quantity* **of type** *int*
**with temporality** $(VT, instant, month)$,
**Attribute** *Amount* **of type** *real*
**with temporality** $(VT, instant, month)$

### 1.5.5 Constraint definitions

*Customer id* **is primary key of** *Customer*;
*Product number* **is primary key of** *Product*;
*Store number* **is primary key of** *Store*;
...
**Snapshot participation of** *Product* **in** *ProdCat* is $(1, 1)$;
**Lifespan participation of** *Category* **in** *ProdCat* is $(1, n)$;
**Snapshot participation of** *Category* **in** *ProdCat* is $(1, 1)$;
**Lifespan participation of** *Category* **in** *ProdCat* is $(1, n)$;
...
**Exclusive participation of** *Customer* **in** *CustSect*, *CustProf*;
**Exclusive participation of** *Branch* **in** *SectBra*, *ProfBra*;

## 1.6 Semantics

In this section we define the semantics of the textual representation of the MultiDim model. We begin by defining the semantics of the predefined data types, as well the model of time. Then, after presenting some auxiliary functions, we give the functions definining the semantics of the different components of the model.

### 1.6.1 Semantics of predefined data types

The semantics of the predefined data types is given by three functions

- A function $\mathcal{D}[\![DATA]\!] \in TF$ such that $\mathcal{D}[\![DATA]\!] : DATA \to SET$. We assume $\forall d \in DATA\ (\perp \in \mathcal{D}[\![DATA]\!](d))$ where $\perp$ represents an undefined value indicating an incorrect use of a function or an error.
- A function $\mathcal{D}[\![OPNS]\!] \in TF$ such that $\mathcal{D}[\![OPNS]\!] : OPNS \to TF$ and $\sigma : d_1 \times \ldots \times d_n \to d$ implies $\mathcal{D}[\![OPNS]\!](\sigma) : \mathcal{D}[\![DATA]\!](d_1) \times \ldots \times \mathcal{D}[\![DATA]\!](d_n) \to \mathcal{D}[\![DATA]\!](d) \in DATA$ for every $d \in DATA$.
- A function $\mathcal{D}[\![PRED]\!] \in TF$ such that $\mathcal{D}[\![PRED]\!] : PRED \to REL$ and $\pi : d_1 \times \ldots \times d_n \to d$ implies $\mathcal{D}[\![PRED]\!](\pi) \subseteq \mathcal{D}[\![DATA]\!](d_1) \times \ldots \times \mathcal{D}[\![DATA]\!](d_n) \to \mathcal{D}[\![DATA]\!](d) \in DATA$ for every $d \in DATA$.

For example, the semantics of the predefined data types and one of their operators are defined as follows:

$$\mathcal{D}[\![DATA]\!](int) = \mathbb{Z} \cup \{\perp\}$$
$$\mathcal{D}[\![DATA]\!](real) = \mathbb{R} \cup \{\perp\}$$
$$\mathcal{D}[\![DATA]\!](string) = \mathbb{A}^* \cup \{\perp\}$$

$$
\begin{aligned}
\mathcal{D}[\![+_i]\!] \ &: \ \mathcal{D}[\![DATA]\!](int) \times \mathcal{D}[\![DATA]\!](int) \rightarrow \mathcal{D}[\![DATA]\!](int) \\
&= \begin{cases} i_1 \times i_2 \rightarrow i_1 + i_2 & \text{if } i_1, i_2 \in \mathbb{Z} \\ \bot & \text{otherwise} \end{cases}
\end{aligned}
$$

### *1.6.2 The time model*

We assume that the real time line is represented in the database by a baseline clock that is discrete and bounded on both ends [1–3]. The time domains are then ordered, finite sets of elements isomorphic to finite subsets of the integer numbers. The non-decomposable elements of the time domain are called *chronons*. Depending on application requirements consecutive chronons can be grouped into a larger unit called a *granule*, such as a second, a minute, or a day. *Granularity* represents the size of the granule, i.e., it is the time unit used for specifying the duration of the granule. We denote a granule as $g$. The granule $g_{now}$ denotes the granule representing current time.

Following Gregersen and Jensen [2], we include a domain for each combination of the temporality types $temp \in \{LS, VT, TT, LT\}$ and granularities $gr$. These domains are denoted $D^{gr}_{temp} = \{g^{temp}_1, g^{temp}_2, \ldots, g^{temp}_n\}$, e.g., $D^{month}_{VT} = \{Jan, Feb, Mar, \ldots, Dec\}$. The domain of each temporal type is the union of domains represented by different granularities: $D_{temp} = \bigcup_{gr}(D^{gr}_{temp})$, e.g., for $TT$ is $D_{TT} = \bigcup_{gr}(D^{gr}_{TT})$.

The real-world instants are represented by a granule according to the chosen granularity, e.g., a granule $g^{day} = 02/10/2006$ using a granularity day. A time interval is defined as the time between two instants called *begin* and *end* instants, i.e., $[g_{begin}, g_{end}]^{gr}$, e.g., $[25/09/2006,\ 02/10/2006]^{day}$. Thus, a time interval is a sequence of consecutive granules between the starting ($g_{begin}$) and ending ($g_{end}$) granules with granularity $gr$, e.g., all days between 25/09/2006 and 02/10/2006. We also denote $Int^{gr}_{temp}$ the interval for each temporality types, e.g., for $VT$ it is $Int^{gr}_{VT}$

We also use set of instants and sets of intervals. An instant set over time domains is a finite union of instants, i.e., $IS^{gr} = g^{gr}_1 \cup \ldots \cup g^{gr}_n$. We denote $IS^{gr}_{temp}$ the instant set for each temporality type *temp*, i.e., for valid time, transaction time, lifespan, and loading time. Furthermore, an interval set, or a temporal element, over time domains is a finite union of intervals, i.e., $TE^{gr} = [g_{begin_1}, g_{end_1}]^{gr} \cup \ldots \cup [g_{begin_n}, g_{end_n}]^{gr}$. We denote $TE^{gr}_{VT}$, $TE^{gr}_{TT}$, $TE^{gr}_{LS}$ and $TE^{gr}_{LT}$ as temporal elements of valid-time, transaction-time, lifespan, and loading time domains, respectively. Since our time domains are discrete and finite, we can define a temporal element as an element of the powerset $\mathcal{P}(D^{gr}_{temp})$.

### 1.6.3 Semantic domains

The MultiDim model includes the following value domains:

$D_S \cup \{\perp\}$ – the set of surrogates
$D_S^L \subseteq D_S$ – the set of surrogates assigned to $L \in \mathit{Levels}$
$D_S^{CP} \subseteq D_S$ – the set of surrogates assigned to $CP \in \mathit{CPRels}$
$D_{LS} = \bigcup_{gr}(D_{LS}^{gr}) \cup \{\perp\}$ – the lifespan domain
$D_{VT} = \bigcup_{gr}(D_{VT}^{gr}) \cup \{\perp\}$ – the valid time domain
$D_{TT} = \bigcup_{gr}(D_{TT}^{gr}) \cup \{\perp\}$ – the transaction time domain
$D_{LT} = \bigcup_{gr}(D_{LT}^{gr}) \cup \{\perp\}$ – the data warehouse loading time domain
$\mathcal{D}[\![DATA]\!]$ – the set of basic domains

### 1.6.4 Auxiliary functions

This section presents auxiliary functions required for defining the semantic functions.

Function *attOf* takes as argument a level declaration or an attribute declaration and returns the attribute names:

$$\begin{aligned}
&\mathit{attOf}(\textbf{Level } L \textbf{ has } A_D) = \\
&\quad \mathit{attOf}(\textbf{Level } L \textbf{ with temporality } T_S \textbf{ has } A_D) = \mathit{attOf}(A_D) \\
&\mathit{attOf}(A_{D_1}, A_{D_2}) = \mathit{attOf}(A_{D_1}) \cup \mathit{attOf}(A_{D_2}) \\
&\mathit{attOf}(\textbf{Attribute } A \textbf{ is } A'_D) = A
\end{aligned}$$

Function *tempOf* takes as argument a temporal specification and returns the temporality types of the specification, i.e., a subset of $\{\mathrm{LS}, \mathrm{VT}, \mathrm{TT}, \mathrm{LT}\}$:

$$\begin{aligned}
&\mathit{tempOf}(T_{S_1}, T_{S_2}) = \mathit{tempOf}(T_{S_1}) \cup \mathit{tempOf}(T_{S_2}) \\
&\mathit{tempOf}((\mathit{temp}, t, gr)) = \{\mathit{temp}\}
\end{aligned}$$

Function *instants* takes as argument a temporal specification and returns a set of functions $t$ (tuples). The domain of each function is the set of temporality types of the specification. The value domain of the temporality types is their underlying temporal domain. Intuitively, the function returns the set of instant tuples of a temporal specification.

$$\begin{aligned}
&\mathit{instants}(T_S) = \{t \mid t \in TF \wedge \mathit{dom}(t) = \{\mathit{tempOf}(T_S)\} \wedge \\
&\quad \forall T_i \in \mathit{tempOf}(T_S)((T_i, t, gr) \in T_S \wedge t[T_i] \in D_{T_i}^{gr}\}
\end{aligned}$$

Function *contains* takes as input an instant tuple $c$ of a temporal specification and a set of instances of a child-parent relationship and returns the subset of the instances that contain $c$ in its temporal support.

$$\mathit{contains}(c, \{t_1, \ldots, t_n\}) =$$

$$\begin{cases} \emptyset & \text{if } n = 0 \\ contains(c, \{t_1, \ldots, t_{n-1}\}) \cup \{t_n\} & \text{if } \forall T_i \in dom(c)(c[T_i] \in t_n[T_i]) \\ contains(c, \{t_1, \ldots, t_{n-1}\}) & \text{otherwise} \end{cases}$$

Function *cnt* takes as input a level member $m$, a level $L$, and a set of instances of a child-parent relationship and returns the number of tuples in the child-parent set in which the member $m$ participates.

$$cnt(m, L, \{t_1, \ldots, t_n\}) = \\ \begin{cases} 0 & \text{if } n = 0 \\ cnt(m, L, \{t_1, \ldots, t_{n-1}\}) & \text{if } n \geq 1 \wedge t_n[s_L] \neq m \\ cnt(m, L, \{t_1, \ldots, t_{n-1}\}) + 1 & \text{if } n \geq 1 \wedge t_n[s_L] = m \end{cases}$$

Function *lifespan* takes as input an identifier of a level member $m$ and a level $L$, and returns the lifespan of the member, if any, or the empty set otherwise.

$$lifespan(m, L) = \quad \begin{cases} t[LS] & \text{if } \exists t \in \mathcal{L}(L)(t[s] = m \wedge LS \in dom(t)\ ) \\ \emptyset & \text{otherwise} \end{cases}$$

Predicate *inSch* takes as first argument a name of a level, of a child-parent relationship, or of a fact relationship, as well as a schema declaration. It returns *true* if the mentioned element is declared in the schema and $false$ otherwise.

$$inSch(L, S_D) = inSch(L, D_D; L_D; CP_D; F_D; IC_D;) = inSch(L, L_D) = \\ \begin{cases} true & \text{if } \textbf{Level } L \textbf{ has } A_D \in L_D \\ true & \text{if } \textbf{Level } L \textbf{ with temporality } T_S \textbf{ has } A_D \in L_D \\ false & \text{otherwise} \end{cases}$$

$$inSch(CP, S_D) = inSch(CP, D_D; L_D; CP_D; F_D; IC_D;) = \\ inSch(CP, CP_D) = \\ \begin{cases} true & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \in L_D \\ true & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\ & \quad \textbf{has distributing factor } \in L_D \\ true & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\ & \quad \textbf{with temporality } T_S \in L_D \\ true & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\ & \quad \textbf{with temporality } T_S \textbf{ has distributing factor } \in L_D \\ false & \text{otherwise} \end{cases}$$

$$inSch(F, S_D) = inSch(F, D_D; L_D; CP_D; F_D; IC_D;) = inSch(F, F_D) = \\ \begin{cases} true & \text{if } \textbf{Fact relationship } F \textbf{ involves } I_S \in L_D \\ true & \text{if } \textbf{Fact relationship } F \textbf{ involves } I_S \textbf{ has } A_D \in L_D \\ false & \text{otherwise} \end{cases}$$

Function *parOf* takes as argument the name of a child-parent relationship and returns the levels that participate in the relationship.

$$
\begin{aligned}
&\mathit{parOf}(CP) = \\
&\begin{cases} \mathit{parOf}(\textbf{C-P relationship } CP \ldots) & \text{if } \textbf{C-P relationship } CP \ldots \in S_D \\ \perp & \text{otherwise} \end{cases} \\
&\mathit{parOf}(\textbf{C-P relationship } CP \textbf{ involves } I_S) = \\
&\qquad \mathit{parOf}(\textbf{C-P relationship } CP \textbf{ involves } I_S \textbf{ has } \ldots) = \\
&\qquad \mathit{parOf}(\textbf{C-P relationship } CP \textbf{ involves } I_S \textbf{ with } \ldots) = \mathit{parOf}(I_S) \\
&\mathit{parOf}(I_{S_1}, I_{S_2}) = \mathit{parOf}(I_{S_1}) \cup \mathit{parOf}(I_{S_2}) \\
&\mathit{parOf}(L) = \{L\}
\end{aligned}
$$

Function *tempSpec* takes as argument the name of a child-parent relationship and returns the specification of its temporal support if the relationship is temporal and the empty set otherwise.

$$
\begin{aligned}
&\mathit{tempSpec}(CP) = \\
&\begin{cases}
\emptyset & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \in E_D \\
\emptyset & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
& \qquad \textbf{has distributing factor} \in E_D \\
T_S & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
& \qquad \textbf{with temporality } T_S \in E_D \\
T_S & \text{if } \textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \\
& \qquad \textbf{with temporality } T_S \textbf{ has distributing factor} \in E_D \\
\perp & \text{otherwise}
\end{cases}
\end{aligned}
$$

Recall that a level may participate several times in a fact relationship using different roles. The functions *role* and *level* takes an involvement of a level in a fact relationship and provides the role name or the level name, respectively.

$$
\begin{aligned}
&\mathit{role}(L) = L \\
&\mathit{role}(L \textbf{ as } \mathit{role}) = \mathit{role} \\
&\mathit{level}(L) = \mathit{level}(L \textbf{ as } \mathit{role}) = L
\end{aligned}
$$

### *1.6.5 Semantic functions*

We give next the signature and the definition of the semantic functions.

The semantic function $\mathcal{I}$ determines the surrogate sets of the levels that are involved in a fact relationship or in a child-parent relationship.

$$
\begin{aligned}
&\mathcal{I}[\![L]\!] : \mathit{Levels} \rightarrow D_S^L \\
&\mathcal{I}[\![I_{S_1}, I_{S_2}]\!] = \mathcal{I}[\![I_{S_1}]\!] \times \mathcal{I}[\![I_{S_2}]\!] \\
&\mathcal{I}[\![L]\!] = \mathcal{I}[\![L \textbf{ as } \mathit{role}]\!] = \begin{cases} D_S^L & \text{if } L \in \mathit{Levels} \\ \perp & \text{otherwise} \end{cases}
\end{aligned}
$$

The semantic function $\mathcal{T}$ determines the time domains of the temporal support specified for a given level, child-parent relationship, or attribute.

$$
\begin{array}{l}
\mathcal{T} : \mathit{Temp\_SPEC} \rightarrow D_{VT} \cup D_{TT} \cup D_{LS} \cup D_{LT} \\
\mathcal{T}[\![\textbf{with temporality } T_{S_1}, T_{S_2}]\!] = \mathcal{T}[\![T_{S_1}]\!] \times \mathcal{T}[\![T_{S_2}]\!] \\
\mathcal{T}[\![(\mathit{temp}, \mathit{instant}, \mathit{gr})]\!] = D^{gr}_{temp} \\
\mathcal{T}[\![(\mathit{temp}, \mathit{instantSet}, \mathit{gr})]\!] = \mathcal{P}(D^{gr}_{temp}) \\
\mathcal{T}[\![(\mathit{temp}, \mathit{interval}, \mathit{gr})]\!] = (\mathit{Int}^{gr}_{temp}) \\
\mathcal{T}[\![(\mathit{temp}, \mathit{intervalSet}, \mathit{gr})]\!] = \mathcal{P}(\mathit{Int}^{gr}_{temp})
\end{array}
$$

The semantic function $\mathcal{A}$ defines the value domains of attribute declarations. If the attribute is of a predefined data type, then the value domain is that of the specified data type. If the attribute of type includes temporal support, this indicates that the value of this attribute changes over time. Therefore, the value domain of this attribute is a function from a time domain to a value domain.

$$
\begin{array}{l}
\mathcal{A} : \mathit{Attributes} \times \mathit{DATA} \times \mathit{Temp\_SPEC} \rightarrow \\
\qquad \mathcal{D}[\![\mathit{DATA}]\!] \cup (\mathcal{T}[\![T_S]\!] \rightarrow \mathcal{D}[\![\mathit{DATA}]\!]) \\
\mathcal{A}[\![A_{D_1}, A_{D_2}]\!] = \mathcal{A}[\![A_{D_1}]\!] \times \mathcal{A}[\![A_{D_2}]\!] \\
\mathcal{A}[\![\textbf{Attribute } A \textbf{ of } A'_D]\!] = \mathcal{A}[\![A'_D]\!] \\
\mathcal{A}[\![\textbf{type } d]\!] = \begin{cases} \mathcal{D}[\![\mathit{DATA}]\!](d) & \text{if } d \in \mathit{DATA} \\ \perp & \text{otherwise} \end{cases} \\
\mathcal{A}[\![\textbf{type } d \textbf{ with temporality } T_S]\!] = \\
\qquad \begin{cases} \mathcal{T}[\![T_S]\!] \rightarrow \mathcal{D}[\![\mathit{DATA}]\!](d) & \text{if } d \in \mathit{DATA} \wedge T_S \in \mathit{Temp\_SPEC} \\ \perp & \text{otherwise} \end{cases}
\end{array}
$$

The function $\mathcal{S}$ defines the semantics of a MultiDim schema composed of definitions of levels, child-parent relationships, fact relationships, and integrity constraints. It defines each component of the underlying database and predicates that ensure validity and consistency of the database.

$$
\begin{array}{l}
\mathcal{S} : S_D \rightarrow \mathcal{S}[\![S_D]\!] \\
\mathcal{S}[\![S_D]\!] = \mathcal{S}[\![L_D; CP_D; F_D; IC_D]\!] \\
\mathcal{S}[\![L_D; CP_D; F_D; IC_D]\!] = \mathcal{L}[\![L_D]\!] \uplus \mathcal{CP}[\![CP_D]\!] \uplus \mathcal{F}[\![F_D]\!] \uplus \mathcal{IC}[\![IC_D]\!]
\end{array}
$$

The function $\mathcal{L}$ defines the semantics of levels. The attributes of a level have an associated value domain. The association between a set of attributes $A = \{A_1, A_2, \ldots, A_n\}$ and the set of value domains $D$ is given by a function $\mathit{dom} : A \rightarrow D$. A member of a level with its attributes can be seen as a tuple. A tuple $t$ over a set of attributes $A$ is actually a function that associates each attribute $A_i \in A$ with a value from the value domain $\mathit{dom}(A_i)$. For an attribute $A$ we denote this value $t[A]$.

The semantics of a level is thus a set of functions $t$ (tuples). The domain of each function $t$ is the surrogate attribute $s$ and the set of attribute names belonging to the level $L$. The value domain of the surrogate attribute $s$ is the set $D^L_S$ of surrogate values assigned to the level $L$ while the value domain of the attributes of the level $L$ is determined by the semantics of the attribute declarations.

If the level has temporal support, this means that the database keeps lifespan, transaction time, and/or loading time for the members of the level. Recall that lifespan indicates the time during which the corresponding real-world member exists, transaction time refers to the time during which the member was current in the database, and loading time refers to the time when a member was introduced in the data warehouse. Therefore, the timestamps recording these temporality types must be associated with the member.

The function $\mathcal{L}$ applied to a composition of level definitions returns the disjoint union of the functions applied to each component. This is because each level defines a unique set of tuples, which is stored separately in the database.

$$
\begin{aligned}
&\mathcal{L} : \mathit{Levels} \times \mathit{Temp\_SPEC} \times \mathit{Att\_DECL} \rightarrow \mathcal{I}[\![L]\!] \times \mathcal{A}[\![A_D]\!] \cup \\
&\qquad \mathcal{I}[\![L]\!] \times \mathcal{T}[\![T_S]\!] \times \mathcal{A}[\![A_D]\!] \\
&\mathcal{L}[\![L_{D_1}; L_{D_2}]\!] = \mathcal{L}[\![L_{D_1}]\!] \uplus \mathcal{L}[\![L_{D_2}]\!] \\
&\mathcal{L}[\![\textbf{Level } L \textbf{ has } A_D]\!] = \\
&\qquad \{t \mid t \in TF \wedge \mathit{dom}(t) = \{s, \mathit{attOf}(A_D)\} \wedge t[s] \in D_S^L \wedge \\
&\qquad \forall A_i \in \mathit{attOf}(A_D)\ (t[A_i] \in \mathcal{A}[\![\textbf{Attribute } A_i \textbf{ of } A'_D]\!])\} \\
&\mathcal{L}[\![\textbf{Level } L \textbf{ with temporality } T_S \textbf{ has } A_D]\!] = \\
&\qquad \{t \mid t \in TF \wedge \mathit{dom}(t) = \{s, \mathit{tempOf}(T_S), \mathit{attOf}(A_D)\} \wedge t[s] \in D_S^L \wedge \\
&\qquad \forall T_i \in \mathit{tempOf}(T_S)(t[T_i] \in \mathcal{T}[\![(T_i, t, gr)]\!]) \wedge \\
&\qquad \forall A_i \in \mathit{attOf}(A_D)(t[A_i] \in \mathcal{A}[\![\textbf{Attribute } A_i \textbf{ of } A'_D]\!])\}
\end{aligned}
$$

The function $\mathcal{CP}$ define the semantics of child-parent relationships. A child-parent relationship relates a child and a parent level and may have in addition temporal support and/or a distributing factor. Their semantics is thus a set of tuples $t$ relating a child and a parent member. Members are identified through their surrogates with the value domain defined by $\mathcal{I}$. If the relationship includes a distributing factor, the domain of the function $t$ includes additionally an attribute $d$; its value domain is the set of real numbers. If the relationship includes temporal support, the timestamps for the different temporality types must be kept.

Since each child-parent relationship defines a unique set of tuples, if the function $\mathcal{CP}$ is applied to a composition of child-relationship relationship definitions, it returns the disjoint union of the functions applied to each component.

$$
\begin{aligned}
&\mathcal{CP} : \mathit{CPRels} \times \mathit{Inv\_SPEC} \times \mathit{Temp\_SPEC} \times \mathit{Attributes} \rightarrow \\
&\qquad \mathcal{I}[\![I_S]\!] \cup \mathcal{I}[\![I_S]\!] \times \mathcal{D}[\![DATA]\!] \cup \mathcal{I}[\![I_S]\!] \times \mathcal{T}[\![T_S]\!] \cup \\
&\qquad \mathcal{I}[\![I_S]\!] \times \mathcal{T}[\![T_S]\!] \times \mathcal{D}[\![DATA]\!] \\
&\mathcal{CP}[\![CP_{D_1}; CP_{D_2}]\!] = \mathcal{CP}[\![CP_{D_1}]\!] \uplus \mathcal{CP}[\![CP_{D_2}]\!] \\
&\mathcal{CP}[\![\textbf{C-P relationship } CP \textbf{ involves } L_1, L_2]\!] = \\
&\qquad \{t \mid t \in TF \wedge \mathit{dom}(t) = \{s_{L_1}, s_{L_2}\} \wedge \\
&\qquad t[s_{L_1}] \in \mathcal{I}[\![L_1]\!] \wedge t[s_{L_2}] \in \mathcal{I}[\![L_2]\!]\}
\end{aligned}
$$

$$\mathcal{CP}[\![\textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \textbf{ has distributing factor}]\!] = \{t \mid t \in TF \wedge dom(t) = \{s_{L_1}, s_{L_2}, d\} \wedge t[s_{L_1}] \in \mathcal{I}[\![L_1]\!] \wedge t[s_{L_2}] \in \mathcal{I}[\![L_2]\!] \wedge t[d] \in \mathcal{D}[\![DATA]\!](real)\}$$

$$\mathcal{CP}[\![\textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \textbf{ with temporality } T_S]\!] = \{t \mid t \in TF \wedge dom(t) = \{s_{L_1}, s_{L_2}, tempOf(T_S)\} \wedge t[s_{L_1}] \in \mathcal{I}[\![L_1]\!] \wedge t[s_{L_2}] \in \mathcal{I}[\![L_2]\!] \wedge \forall T_i \in tempOf(T_S)(t[T_i] \in \mathcal{T}[\![(T_i, t, gr)]\!]) \wedge (LS \in tempOf(T_S) \Rightarrow t[LS] \subseteq lifespan(s_{L_1}, L_1) \cap lifespan(s_{L_2}, L_2))\}$$

$$\mathcal{CP}[\![\textbf{C-P relationship } CP \textbf{ involves } L_1, L_2 \textbf{ with temporality } T_S \textbf{ has distributing factor}]\!] = \{t \mid t \in TF \wedge dom(t) = \{s_{L_1}, s_{L_2}, tempOf(T_S), d\} \wedge t[s_{L_1}] \in \mathcal{I}[\![L_1]\!] \wedge t[s_{L_2}] \in \mathcal{I}[\![L_2]\!] \wedge \forall T_i \in tempOf(T_S)(t[T_i] \in \mathcal{T}[\![(T_i, t, gr)]\!]) \wedge (LS \in tempOf(T_S) \Rightarrow t[LS] \subseteq lifespan(s_{L_1}, L_1) \cap lifespan(s_{L_2}, L_2)) \wedge t[d] \in \mathcal{D}[\![DATA]\!](real)\}$$

Notice that the two last definitions above enforce the constraint that the lifespan of an instance of a temporal child-parent relationship must be included in the intersection of lifespans of its participating members.

The function $\mathcal{F}$ defines the semantics of fact relationships. A fact relationship relates several levels and may have attributes. Its semantics is thus a set of tuples $t$ defining a member from each of its levels, as well as values for its attributes. Recall that a level may participate several times in a fact relationship using different roles. If this is the case the role name is used instead of the level name in the domain of function $t$. Members are identified through their surrogates with the value domain defined by $\mathcal{I}$. If the fact relationship has attributes, the domain of the function $t$ includes additionally the set of attribute names. The value domains of these attributes are determined by the semantics of the attribute declarations. As for the level and the child-parent relationships, a fact relationship defines a unique set of tuples that are stored separately in the database.

$$\mathcal{F} : FactRels \times Inv\_SPEC \times Att\_DECL \rightarrow \mathcal{I}[\![I_S]\!] \cup \mathcal{I}[\![I_S]\!] \times \mathcal{A}[\![A_D]\!]$$

$$\mathcal{F}[\![F_{D_1}; F_{D_2}]\!] = \mathcal{F}[\![F_{D_1}]\!] \uplus \mathcal{F}[\![F_{D_2}]\!]$$

$$\mathcal{F}[\![\textbf{Fact relationship } F \textbf{ involves } I_S]\!] = \{t \mid t \in TF \wedge dom(t) = \{\textstyle\bigcup_{L_i \in I_S} s_{role(L_i)}\} \wedge \forall L_i \in I_S\ (t[s_{role(L_i)}] \in \mathcal{I}[\![level(L_i)]\!])\}$$

$$\mathcal{F}[\![\textbf{Fact relationship } F \textbf{ involves } I_S \textbf{ has } A_D]\!] = \{t \mid t \in TF \wedge dom(t) = \{\textstyle\bigcup_{L_i \in I_S} s_{role(L_i)}, attOf(A_D)\} \wedge \forall A_i \in attOf(A_D)\ (t[A_i] \in \mathcal{A}[\![\textbf{Attribute } A \textbf{ of } A'_D]\!]) \wedge \forall L_i \in I_S\ (t[s_{role(L_i)}] \in \mathcal{I}[\![level(L_i)]\!])\}$$

The function $\mathcal{IC}$ defines the semantics of the integrity constraints. The semantics of a constraint is a set of predicates that the database must satisfy. In the textual representation all constraints are separate constructs, so the predicates must first verify that the constructs (e.g, levels, relationships) mentioned in the constraints belongs to the schema using the function *inSch*.

$$\mathcal{IC} : IC_D \rightarrow PRED$$
$$\mathcal{IC}[\![IC_{D_1}; IC_{D_2}]\!] = \mathcal{IC}[\![IC_{D_1}]\!] \wedge \mathcal{IC}[\![IC_{D_2}]\!]$$

The primary key constraint ensures that the values of the key attributes are unique for all members of the level.

$$\begin{aligned}
&\mathcal{IC}[\![K \textbf{ is primary key of } L]\!] = \mathit{inSch}(L, S_D) \wedge \\
&\quad ((K \subseteq \mathit{attOf}(\textbf{Level } L \textbf{ has } A_D) \wedge \forall t_i, t_j \in \mathcal{L}[\![\textbf{Level } L \textbf{ has } A_D]\!] \\
&\quad (t_i[K] = t_j[K] \Rightarrow t_i[s] = t_j[s])) \vee \\
&\quad (K \subseteq \mathit{attOf}(\textbf{Level } L \textbf{ with temporality } T_S \textbf{ has } A_D) \wedge \\
&\quad \forall t_i, t_j \in \mathcal{L}[\![\textbf{Level } L \textbf{ with temporality } T_S \textbf{ has } A_D]\!] \wedge \\
&\quad (\mathcal{T}[\![T_S]\!] \rightarrow t_i[K] = \mathcal{T}[\![T_S]\!] \rightarrow t_j[K] \Rightarrow \mathcal{T}[\![T_S]\!] \rightarrow t_i[s] = \mathcal{T}[\![T_S]\!] \rightarrow t_j[s]))
\end{aligned}$$

The cardinality constraints ensure that a child member can be related to minimum *min* and maximum *max* parent members. Three cases must be considered: usual cardinality constraints for non-temporal relationships as well as snapshot and lifespan cardinality constraints for temporal relationships. In the case of usual cardinality constraints, for every member $m$ of the level $L$ we use the function *cnt* to determine the number of tuples belonging to the semantics of the child-parent relationship in which $m$ participates. Snapshot cardinality constraints must be satisfied at each instant tuple $c$ of the temporal domain of the relationship. Therefore, we use the function *contains* to obtain the subset of the tuples belonging to the semantics of the child-parent relationship that contains the instant tuple $c$. Then the function *cnt* is applied to this subset. In addition, if the level participating in the relationship is temporal, then for each instant belonging to the lifespan of a member $m$, it must be related to a valid member of the other level through an instance of the relationship. Finally, lifespan cardinality constraints must be satisfied during the whole temporal domain of the relationship.

$$\begin{aligned}
&\mathcal{IC}[\![\textbf{Participation of } L \textbf{ in } \mathit{CP} \textbf{ is } (\mathit{min}, \mathit{max})]\!] = \\
&\quad \mathit{inSch}(L, S_D) \wedge \mathit{inSch}(\mathit{CP}, S_D) \wedge L \in \mathit{parOf}(\mathit{CP}) \wedge \forall m \in D_S^L \\
&\quad (\mathit{min} \leq \mathit{cnt}(m, L, \mathcal{CP}[\![\textbf{C-P relationship } \mathit{CP} \ldots]\!]) \leq \mathit{max}) \\
&\mathcal{IC}[\![\textbf{Snapshot participation of } L \textbf{ in } \mathit{CP} \textbf{ is } (\mathit{min}, \mathit{max})]\!] = \\
&\quad \mathit{inSch}(L, S_D) \wedge \mathit{inSch}(\mathit{CP}, S_D) \wedge L \in \mathit{parOf}(\mathit{CP}) \wedge \\
&\quad \forall c \in \mathit{instants}(\mathit{tempSpec}(\mathit{CP}))\ \forall m \in D_S^L(\mathit{min} \leq \\
&\quad \mathit{cnt}(m, L, \mathit{contains}(c, \mathcal{CP}[\![\textbf{C-P relationship } \mathit{CP} \ldots]\!])) \leq \mathit{max}) \wedge \\
&\quad \forall m \in D_S^L\ \forall c \in \mathit{lifespan}(m, L)\ \exists t \in \mathcal{CP}[\![\textbf{C-P relationship } \mathit{CP} \ldots]\!] \\
&\quad (t[s_L] = m \wedge c \in t[LS]) \\
&\mathcal{IC}[\![\textbf{Lifespan participation of } L \textbf{ in } \mathit{CP} \textbf{ is } (\mathit{min}, \mathit{max})]\!] = \\
&\quad \mathit{inSch}(L, S_D) \wedge \mathit{inSch}(\mathit{CP}, S_D) \wedge L \in \mathit{parOf}(\mathit{CP}) \wedge \forall m \in D_S^L \\
&\quad (\mathit{min} \leq \mathit{cnt}(m, L, \mathcal{CP}[\![\textbf{C-P relationship } \mathit{CP} \ldots]\!]) \leq \mathit{max})
\end{aligned}$$

A member of a level that participates exclusively in a set of child-parent relationships (i.e., a member of a splitting or joining level) cannot be involved in more than one of these relationships.

$$\begin{aligned}
&\mathcal{IC}[\![\textbf{Exclusive participation of } L \textbf{ in } CP_S]\!] = inSch(L, S_D) \wedge \\
&\quad \forall CP_i \in CP_S(inSch(CP_i, S_D)) \wedge \neg(\exists CP_i, CP_j \in CP_S \\
&\quad \exists t_1 \in \mathcal{CP}[\![\textbf{C-P relationship } CP_i \textbf{ involves } \ldots]\!] \\
&\quad \exists t_2 \in \mathcal{CP}[\![\textbf{C-P relationship } CP_j \textbf{ involves } \ldots]\!] \\
&\quad (i \neq j \wedge t_1[s_L] = t_2[s_L]))
\end{aligned}$$

Notice that in the above formalization dimensions and hierarchies do not have semantic interpretations. However, they are needed for defining meaningful OLAP operations. Dimensions are required for the drill-across operation that allows to compare measures from different fact relationships. Hierarchies are needed for defining aggregations for the roll-up and drill-down operations. Such operations are beyond the scope of this paper.

## References


[1] O. Etzion, S. Jajodia, and S. Sripada, editors. *Temporal Databases: Research and Practice*. Springer, 1998.

[2] H. Gregersen and C. Jensen. Conceptual modeling of time-varying information. Technical report, Time Center, TR-35, 1998.

[3] C. Jensen and R. Snodgrass. Temporally enhanced database design. In M. Papazoglou, S. Spaccapietra, and Z. Tari, editors, *Advances in Object-Oriented Data Modeling*, pages 163–193. MIT Press, 2000.