A Conceptual Model for Multidimensional Data

Anand S. Kamble

Department of Information Technology
Government of India
New Delhi, India
Email: ask@mit.gov.in

Abstract

This paper introduces a Conceptual Data Model for Data Warehouse including multidimensional aggregation. It is based on Entity-Relationships data model. The conceptual data model gracefully extends standard Entity-Relationship data model with multidimensional aggregated entities. The model has a clear mathematical theoretic semantics grounded on standard ER semantics and the $\mathcal{GMD}$ logic-based multidimensional data model. The aim of this work is not to propose yet another conceptual data model, but to find the most general and precise formalism considering all the proposals for a conceptual data model in the data warehouse field, making therefore a possible formal comparison of the differences of the models in the literature, and to study the formal properties or extensions of such data models.

1 Introduction

The goal of this work is to extend the standard Entity-Relationship (ER) data model, as defined in the database textbooks, with constructs which allow the modeling of multidimensional aggregated entities together with their interrelationships with the other parts of the conceptual schema. An important aspect is that a formal model-theoretic semantics is to be given to the conceptual data model by combining the well known first order semantics of standard ER, as described for example, in (Borgida et al. 2003, Calvanese et al. 1998)—with the model theoretic semantics of the $\mathcal{GMD}$ logical multidimensional data model (Franconi & Kamble 2003, 2004*b*). This work is also based on a similar preliminary work done on the use of Description Logics as a mean to give precise semantics to a data warehouse conceptual data model and to study its computational properties (Franconi & Sattler 1999). This paper presents the formal aspects along with well defined model-theoretic syntax and semantics of the conceptual data model introduced in (Franconi & Kamble 2004*a*).

The proposed framework is a novel data warehouse conceptual data model, $\mathcal{CGMD}$–generalising conceptual multidimensional data models in the data warehouse field. The aim of this work is not to propose yet another data model, but to find the most general, an elegant and precise formalism encompassing all the proposals, for example, listed in (Phipps & Davis 2002), for a conceptual data model in the data warehouse field, making therefore a possible formal comparison of the different expressivities of the models in the literature.

The paper is organised as follows. Section 2 describes the $\mathcal{CGMD}$ model along with the required extensions to the standard entity-relationships data model. In Section 3 and Section 4, we present respectively syntax and semantics of the $\mathcal{CGMD}$ model, which are purely based on mathematical theory. Section 5 reviews the related literature on multidimensional conceptual models. Section 6 evaluates the $\mathcal{CGMD}$ model against the criteria for a good conceptual multidimensional model. In Section 6, we present comparison of the $\mathcal{CGMD}$ model with other multidimensional models. Finally, in Section 8, we briefly conclude the paper and outline the future work.

2 The $\mathcal{CGMD}$ Data Model

The $\mathcal{CGMD}$ model extends ideas of a data warehouse conceptual data model first proposed in (Franconi & Sattler 1999) where aggregations and dimensions are first class citizens. It abstracts principles of data warehouse and describes the multidimensional structure of the data of a business domain of an enterprise.

A $\mathcal{CGMD}$ model is based on an ER model. It captures database schemata expressed in an entity-relationship diagram and describes multidimensional structure including dimensions with their hierarchically organised levels and the structure of aggregations. It extends standard ER schema with constructs of aggregated entities together with their interrelationships with the other parts of the schema. As stated in (Agrawal et al. 1997), a "good" data warehouse system should support user-definable *multiple* hierarchies along the *arbitrary* dimensions. A $\mathcal{CGMD}$ model is able to support user-definable multiple hierarchies, and is able to express aggregations along the arbitrary dimensions and levels.

2.1 The $\mathcal{CGMD}$: an Extended Entity-Relationship Model

This section describes the $\mathcal{CGMD}$ data model with an ER model and presents the ER extensions. It also presents methodology for data warehouse design from the standard (operational) ER schema and the structure of aggregations.

We describe the model with an example of telephone calls presented in Figure 1 (taken from (Franconi & Sattler 1999)). Entities `Calls`, `Day` and `Point`, and relationships such as `Date, Dest` and `Source` present the base data. The cardinality constraints such as (1,1) on the "Date" relationship between "Calls" & "Day" entities, and the "Source"/"Dest" (Destination) relationship between "Calls" & "Point" entities express that the calls are issued on some dates from some source points and receiving at some destination

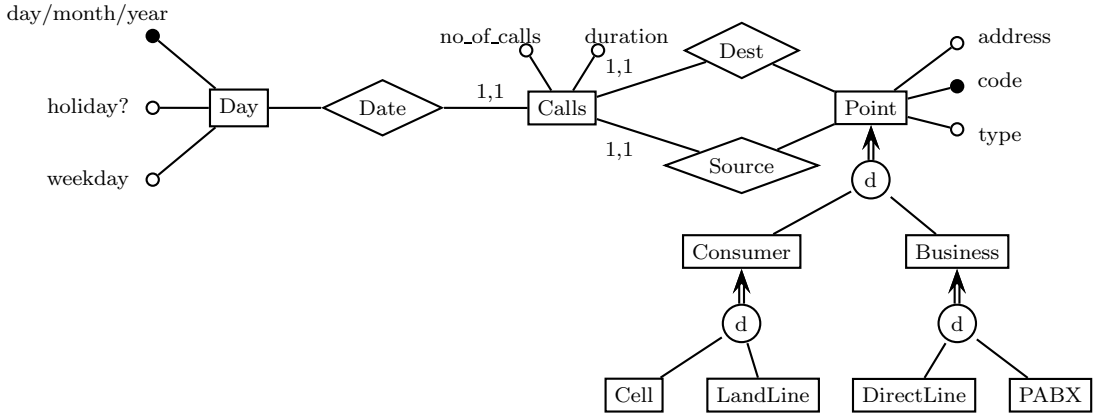

Figure 1: The Conceptual Schema for the base data of telephone calls

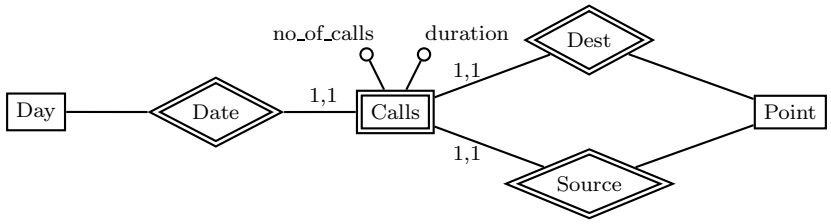

Figure 2: The Conceptual Schema for the Basic Multidimensional information for the base data considered in Figure 1

points. The conceptual multidimensional data model for this information (base data) we have obtained, is exemplified in Figure 2. A basic multidimensional entity such as `Calls` described in the diagram of the figure 2 is using a standard star schema—i.e., it is represented by means of a weak entity with respect to its dimensions. In this example, this basic multidimensional entity may be useful for analysing the nature of telephone calls by considering, among others, the dimension related to the origin and the destination of the calls with respect to the type of phone point (associated to consumer or business customers). So, the entity `Calls` represents a basic cube whose dimensions are `Date, Dest` and `Source` (identifying relationships) which are restricted to the basic levels `Day, Point`, and gain `Point` (associated entities) respectively. This part of the diagram makes still use of standard constructs.

Level building: For building the aggregation level hierarchies for each dimension, we consider the following:

- discriminator of an entity (Elmasri & Navathe 2000)
- generalisation/specialisation hierarchy: creating a single entity of all subclasses (possibly disjoint) of a superclass.
- one-to-many relationship
- partial relationship
- many-to-many relationship: converting into one-to-many relationships which are then converted into levels as suggested in (Moody & Kortink 2000).

Taking these constraints into account, Figure 3 presents the multidimensional conceptual schema including level hierarchies for each dimension. Outer boxes indicate levels; and inner boxes are their elements. The bold arrows (from lower level to higher level) denote hierarchy. The levels "Pointtype" and "Customertype" are created from the partitions of "Point" entity. An entity Point is partitioned (according to an attribute "type"—a discriminator (Elmasri & Navathe 2000)) into four basic points and two higher level points. Pointtype aggregates four basic point types (partitions) namely, `Cell, Land Line, Direct Line` & `PABX`, and Customertype aggregates higher level two (partitions) points types (partitions) viz `Consumer` and `Business`. Thus, first three constraints namely, discriminator, generalisation/secialisation, one-to-many relationships hold. Similarly, Date dimension multiple hierarchies (for instance, hierarchy including the levels DAY, Month, Qtr, Year) are created. The Holiday-Nonholiday level aggregates all holidays and non-holidays. However, in this case, Holiday or Nonholiday is an optional level as the relationship between Day and Holiday or Nonholiday is partial, hence, partial relationships. In running example, we do not have many-to-many relationships, however, handling them is straightforward as suggested in (Moody & Kortink 2000).

Aggregation:

We now perform the analysis of telephone calls along the arbitrary dimensions. For example, a query "Analyse telephone calls by day and point type?" is a bi-dimensional cube along the Date and the Source dimensions involving the level Day and the level Pointtype respectively. A conceptual schema for this query includes the definition of the basic cube (Figure 2) and the definition of the aggregation along the definitions of associated levels, i.e., a new aggregated entity `Calls-by-Day-and-Pointtype` denoting aggregations according to the basic level `Day` and the level `Pointtype` along the dimensions `Date` and `Source` respectively. Figure 4 presents the conceptual schema

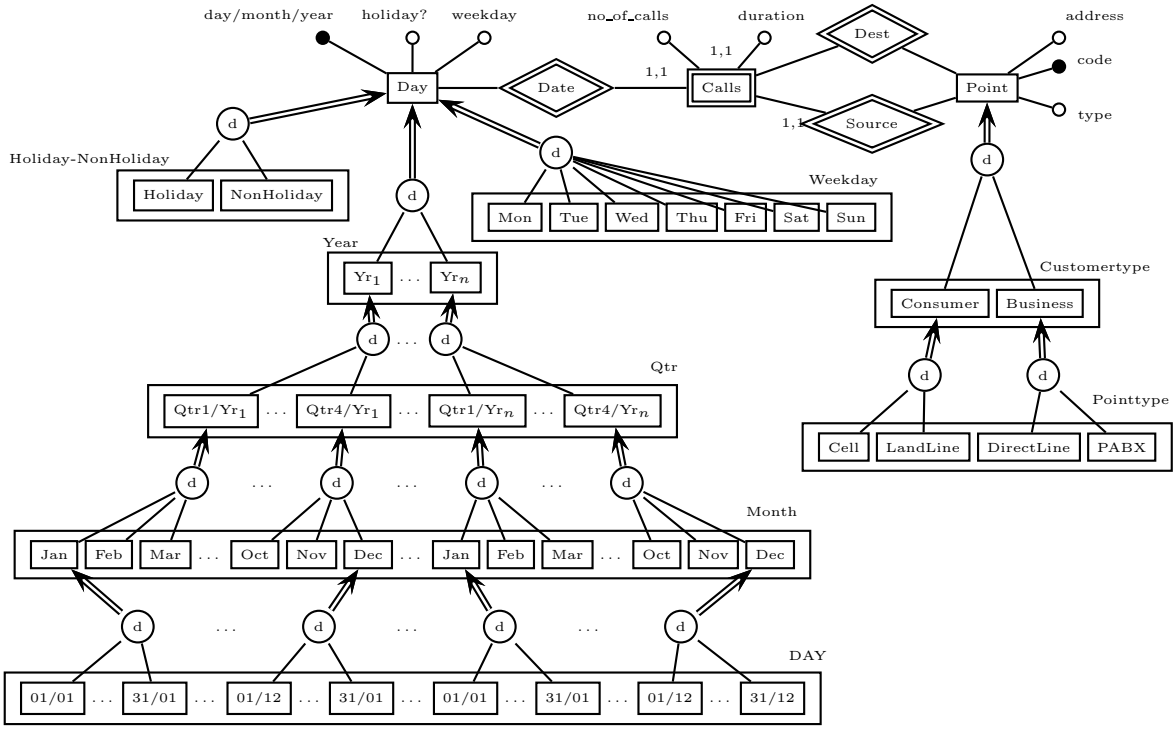

Figure 3: The Multidimensional Conceptual Schema for the data considered in Figure 1.

for this aggregated cube (query) in the variant of an Entity-Relationship model. This particular way of presenting aggregation (entity) is adapted from UML (Unified Modeling Language) syntax.

Now consider a *multidimensional aggregated view*, for example, "analysis of telephone calls by week day and customer type", composing telephone calls along the `Date` and the `Source` dimensions involving levels `Weekday` and `Customertype` respectively. The conceptual schema for this aggregated view includes the definition of the basic cube and the definition of aggregation, i.e., a new aggregated entity, say `Calls-by-Weekday-and-Customertype` (along with the definitions of the level `Weekday` and the level `Customertype`) denoting aggregations according to the level `Weekday` and the level `Customertype` along the `Date` and the `Source` dimensions respectively. Figure 5 presents the conceptual schema in the variant of an ER model. This bi-dimensional aggregated view is actually computed from an aggregated cube of Figure 4. This indicates that the aggregations can be computed from pre-computed aggregations.

2.2 Extensions to the ER Model

As described above, a **first** extension to the standard ER Model can be seen with `simple aggregated entities`—i.e., non-dimensional aggregations—`Weekday` and `Customertype`, which represent dimensional levels built from the basic dimensional entities `Day` and `Point` respectively. A simple aggregation aggregates the collection of objects that are in the extension of the aggregated entities. So, in our example, since entities `Mon,...,Sun` form a partition of the entity `Day`, the `Weekday` entity denotes exactly seven objects, one for all the Mondays, one for all the Tuesdays, etc. On the other hand, the aggregated entity `Customertype` denotes exactly two objects, one aggregating all customer phone points and the other aggregating all business phone points. In this, by interleaving partitioning and simple aggregations, we are able to construct `level hierarchies` starting from some basic dimensional level. Obviously, the functional dependencies exist among the levels of a hierarchy, as analysed by (Golfarelli et al. 1998).

A **second** extension to the standard ER model is the `multidimensional aggregated entity` exemplified in Figure 5 by the entity `Calls-by-Weekday-and-Customertype` and in Figure 4 by the entity `Calls-by-Day-and-Pointtype`. The entity `Calls-by-Weekday-and-Customertype` denotes all the cells of a cube whose coordinates are the weekdays of the date of the calls, and the customer types of the originators of the calls. Such an entity (i.e., extension) holds the necessary constraints enforced for a cube by the $\mathcal{GMD}$-based semantics (Franconi & Kamble 2003, 2004*b*).

A multidimensional aggregated entity is an entity itself in the ER diagram, and it can have attributes (for instance, total_no_of_calls and average_duration in Figure 5 or Figure 4, and can be computed with associated aggregation functions, i.e., `sum(no_of_calls)` and `average(duration)` respectively) and can be part of further relationships or constraints.

3 Syntax of the $\mathcal{CGMD}$ Data Model

The basic constructs of the ER schema are entities, relationships. and attributes. Entity is drawn as a rectangle around the entity symbol (entity name), whereas relationship between the entities is drawn as a diamond around the symbol (relationship name). An attribute is drawn as a circle or oval outside or around the attribute symbol (attribute name). ER-roles are the edges (links) between entities and relationships and are labeled with number restrictions called cardinality constraints. An *is-a* link constraint is drawn as an arrow from more specific entity (subclass) to more general entity called superclass (respectively from more specific to more general relationship). The *disjoint-total* constraint is drawn with a

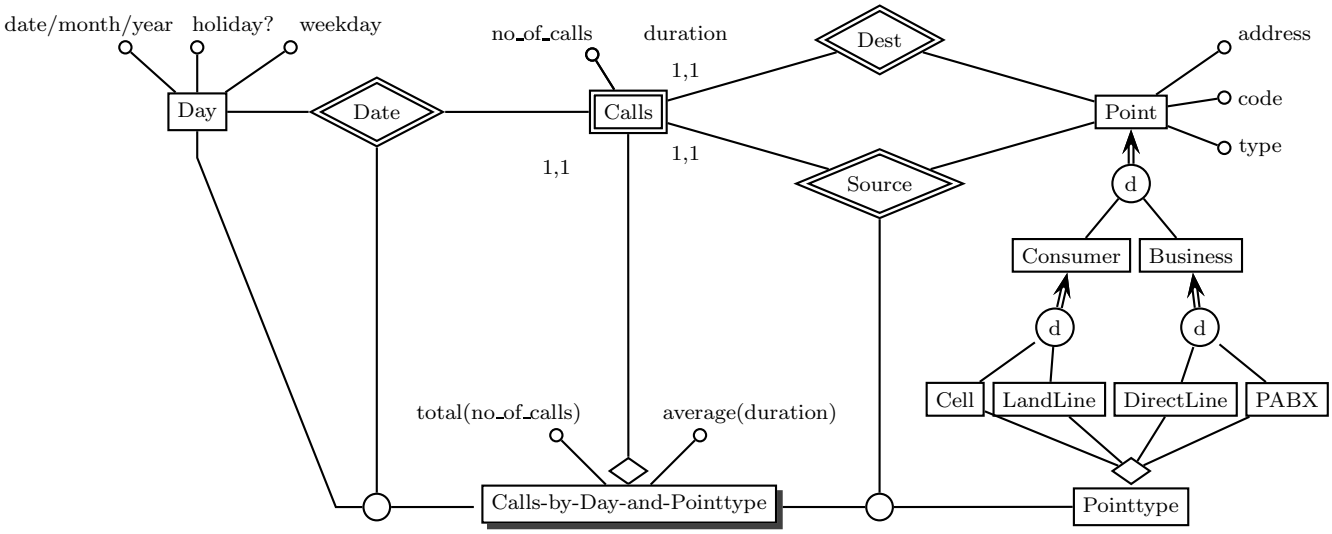

Figure 4: A cube composing calls by level Day and level Pointtype along Date and Source dimensions respectively

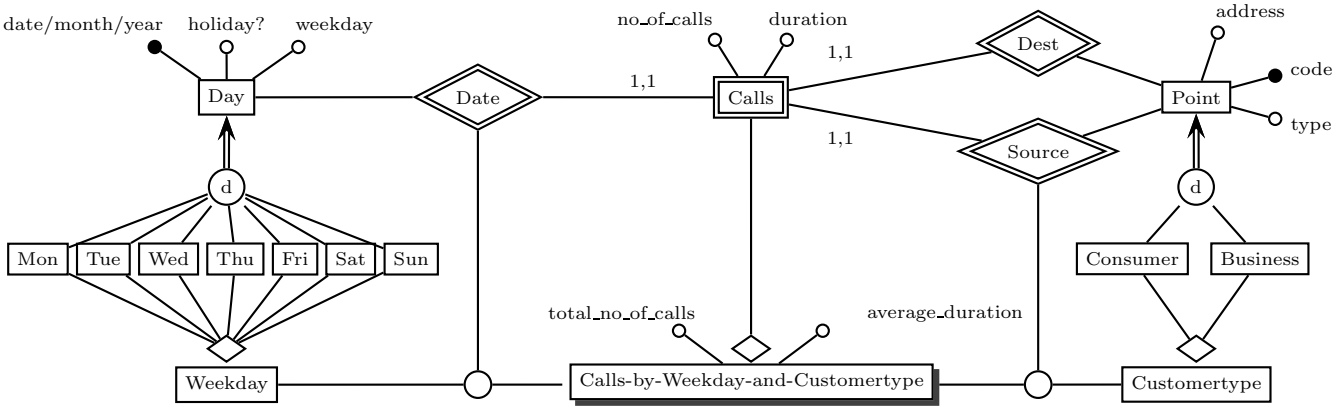

Figure 5: The Conceptual Data Warehouse Schema for multidimensional aggregated view presenting Calls by Weekday and Customertype—an aggregated cube (view)

circle having "d" inside, connecting subclasses with the edges and a superclass with a double-lined arrow from the circle to a superclass. Weak entities are represented as double-lined rectangles, whereas identifying relationships are denoted by double-lined diamonds. Aggregated entity (simple aggregation) is drawn as rectangle attached with diamond, whereas multidimensional aggregated entity is drawn as shadowed rectangle attached with diamond.

Formally, the syntax of an Extended Entity-Relationship (EER) model is as follows.

Definition 1 (EER schema) *An EER schema is constructed over the signature* $\mathcal{S} = < \mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{U}, \mathcal{V}, \preceq, card, \{\!|\cdot|\!\} >$ *where*

- $\mathcal{E}$ *is a finite set of entity names,*
- $\mathcal{R}$ *is a finite set of relationship names, each associated with an arity k,*
- $\mathcal{A}$ *is a finite set of attribute names,*
- $\mathcal{U}$ *is a finite set of ER-role names,*
- $\mathcal{V}$ *is a finite set of domain names,*
- $\preceq \subseteq \mathcal{E} \times \mathcal{E} \cup \mathcal{R} \times \mathcal{R}$ *is a binary relation over* $\mathcal{E}$ *and* $\mathcal{R}$
- *card is a function such that* $card(E,R,U) = < cmin(E,R,U), cmax(U) > \in \mathbb{N} \times \mathbb{N}$ *where* $cmin(E,R,U) \leq cmax(E,R,U)$ *for each* $E \in \mathcal{E}$, $R \in \mathcal{R}$, $U \in \mathcal{U}$.
- $\{\!|\cdot|\!\}$ *models aggregation over* $\mathcal{E}$.

An EER schema $\mathbb{E}$ *over the signature* $\mathcal{S}$ *is*

- *a finite set of entities* $E \in \mathcal{E}$,
- *a finite set of relationship constraints* R *of an arity* k *such that* $R \doteq [U_1 : E_1, \ldots, U_k : E_k]$ *where* $R \in \mathcal{R}$, $E_i \in \mathcal{E}$ *and* $U_i \in \mathcal{U}$ *for each* i, $1 \leq i \leq k$,
- *a finite set of attribute constraints* $A_i \in \mathcal{A}$ *such that* $E \doteq \{A_1 : V_1, \ldots, A_n : V_n\}$ *where* $A_i \in \mathcal{A}$, $V_i \in \mathcal{V}$, $E \in \mathcal{E}$ *is in* $\mathbb{E}$ *for each* i, $1 \leq i \leq n$,
- *a finite set of* is-a *link constraints between two entities* E_1 *and* E_2 *such that* $E_1 \preceq E_2$, *(respectively between two relationships* R, S *such that* $R \preceq S$*),*
- *a finite set of* disjoint-total *constraints between more specific entities* $E_1 \ldots E_n$ *and a more general entity* E *such that* $E_1 \preceq E$, $\ldots$, $E_n \preceq E$ *where* $E_i \neq E_j$ *for* $i \neq j, i \leq n; j \leq n$, E, $E_k \in \mathcal{E}$ *for each* k, $k = 1, \ldots, n$

- *a finite set of aggregation links between entities, each aggregation link in* $\mathbb{E}$ *is drawn as an edge with attached diamond at one end*
- *a finite set of simple aggregations* $G \in \mathcal{E}$ *involving* n *entities* $F_1, \ldots, F_n$ *(each being connected with an aggregation link to G)*
- *a finite set of aggregations G involving (connecting)* n *relations* $D_1, \ldots, D_n$ *and* n *entities* $L_1, \ldots, L_n$ *and a weak entity F.*

Before giving the formal semantics of the EER model, we describe intuitively the components of the EER Schema. An entity denotes a set of objects called instances that have common properties. The elementary properties are modelled with attributes whose values belong to one of several predefined domains such as Integer, Real, String, or Boolean. The properties that are due to relations to other entities are modelled through the participation of the entities in the relationships. A relationship denotes a set of tuples called its instances, each of which represents an association among different combination of instances of the entities that participate in the relationship. Each entity can participate in a relationship more than once. Such participation is represented by an ER-role. Each ER-role is assigned a unique name. Number of ER-roles associated to a relationship is called the arity of that relationship. The cardinality constraints (number restrictions) are associated to ER-role in order to restrict the number of times participation of each instance of an entity via that ER-role in instances of the relationship. The minimum cardinality is either 0 (zero) or 1 (one) and maximum cardinality is either 1 or ∞. An *is-a* link is modelled by $\preceq$ and is used to denote the inclusion between two entities (respectively between two relationships) and therefore more specific entity (respectively relationship) inherits properties of more general entity (respectively relationship).

Weak entity is a dependent entity which is identified by considering the primary keys of participation of other entities via relationships (called weak relationships) to which it is connected via ER-roles, each having minimum and maximum cardinalities equal to 1. Each instance of weak entity is a composition of instances of participating entities (one instance per entity).

A fact is represented as weak entity (aggregated fact as aggregated weak entity). The dimensions are represented as relationships (weak relationships) and levels are represented as entities (also including aggregated entities). An entity (level) directly connected to a (weak) relationship (dimension) is called a basic level for that dimension. A roll-up link between two levels (entities) is modeled by a roll-up function ρ which maps lower level elements (instances) to higher level elements (instances). Simple aggregation is represented by an aggregated entity which is a composition of entities (to which it is connected with lines). i.e., Simple aggregation involves a finite set of entities on which it is based on. An n-dimensional aggregation is represented by an aggregated weak entity connecting n relationships (dimensions), n entities (levels) by connecting them via n circles (one per relationship and per entity), and a weak entity (fact on which aggregation is based on) connecting to it (with line), i.e., an n-dimensional aggregation is represented by an aggregation (aggregated weak entity) involving n dimensions (each being a relationship), n levels (each being an entity or aggregated entity) and a fact (weak entity) on which aggregation is based on. That is an n-dimensional aggregation involves n dimensions, n levels (one per dimension) and a fact it is based on. Each instance of n-dimensional aggregation is called a cell which is a composition of n elements (instances) of n levels (one element/instance per level) of n dimensions involved in the aggregation. In rest of the paper, we will many times use only "aggregation" to refer multidimensional or n-dimensional aggregation.

4 Semantics of the $\mathcal{CGMD}$ Data Model

The semantics of an EER Schema is given in terms of legal data warehouse states, i.e., data warehouses which conform to the constraints imposed by the schema. We consider as a starting point the ER semantics introduced in (Calvanese et al. 1998), recasted to cope with multidimensional information. For we consider $\mathcal{GMD}$, the logical multidimensional data model introduced in (Franconi & Kamble 2003). $\mathcal{GMD}$ abstracts notions such as levels, multiple level hierarchies, dimensions, facts, cells, aggregation, cube, coordinates and measures. A central element in $\mathcal{GMD}$ is a cube. A cube defined on all specified dimensions with their basic levels is called a basic cube, otherwise, it is called an aggregated cube. A cube is computed from a cube. An aggregated cube is computed from a cube on which the aggregation is based on. The $\mathcal{GMD}$ introduces a notion of data warehouse state. A data warehouse state is a collection of cells (with their dimensions and measures). A data warehouse state is legal if it satisfies the above cube conditions.

Definition 2 (EER Semantics) *A data warehouse state* $I = < \Delta, \Gamma, \cdot^I >$ *over the signature* $< \mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{U}, \mathcal{V}, \preceq, card, \{\!|\cdot|\!\} >$ *with respect to the EER schema* $\mathbb{E}$ *is constituted by*

- Δ *a nonempty finite set assumed to be different from all domains,*
- Γ *a finite set of (concrete) domains*
- $\cdot^I$ *an interpretation function such that*
  - $V^I \subseteq \Gamma$ *for each* $V \in \mathcal{V}$*, where* V^I *is disjoint from any other* W^I *such that* $W \in \mathcal{V}$
  - $E^I \subseteq \Delta$ *for each* $E \in \mathcal{E}$*, where* E^I *is disjoint from any other* E'^I *such that* $E' \in \mathcal{E}$
  - $A^I \subseteq \Delta \times V^I$ *for each* $A \in \mathcal{A}$*, and for some* $V \in \mathcal{V}$
  - $R^I \subseteq \Delta \times \ldots \times \Delta = \Delta^k$ *for each* k*-ary relationship* $R \in \mathcal{R}$ *such that a tuple* $r \in R^I$ *is of the form* $[U_1{:}e_1, \ldots, U_k{:}e_k]$*, where* $e_i \in E_i^I$*, for each* $i \in \{1, \ldots, k\}$.

A tuple $r \in R^I$ *over* Δ *can be viewed as a function that maps each ER-role* U_i *to* $e_i \in E_i$ *and is denoted by* $[U_1{:}e_1, \ldots, U_k{:}e_k]$*, i.e.,* $r[U_i] = e_i \in E_i$ *for each* i*,* $i = 1, \ldots, k$.

The elements of E^I, A^I*, and* R^I *are called instances of E, A, and R, respectively.*

A data warehouse state $I = < \Delta, \Gamma, \cdot^I >$ *is said to be* legal *for an EER schema* $\mathbb{E}$*, if it satisfies the following:*

- $E_1^I \subseteq E_2^I$ *for each* is-a *link in* $\mathbb{E}$ *between two entities* E_1, E_2 *in* $\mathbb{E}$ *such that* $E_1 \preceq E_2$

  Similarly $R_1^I \subseteq R_2^I$ *for each* is-a *link between relationships* R_1, R_2 *i* $\mathbb{E}$ *such that* $R_1 \preceq R_2$

- $A^I(e) \in V^I$ *for each* $e \in E^I$, *where* $A \in \mathcal{A}$ *is an attribute of* E *with domain* $V \in \mathcal{V}$.

  Similarly, $A^I(r) \in V^I$ *for each* $r \in R^I$, *where* $A \in \mathcal{A}$ *is an attribute of* R *with domain* $V \in \mathcal{V}$

- $R^I \subseteq E_1^I \times \ldots \times E_k^I$ *for each relationship* R *in* $\mathbb{E}$ *connected to entities* $E_1, \ldots, E_k$ *in* $\mathbb{E}$

- $cmin(E,R,U) \leq \#\{r \in R^I \mid r[U] = e\} \leq cmax(E,R,U)$
  for each $U \in \mathcal{U}$, *associated to* $R \in \mathcal{R}$ *and* $E \in \mathcal{E}$ *in* $\mathbb{E}$, *for each* $e \in E^I$, *and cardinality constraint* $card(U) = (min, max)$ *associated with ER-role* U *where* $cmin(E,R,U) = min$ *and* $cmin(E,R,U) = max$

- *for each disjoint-total construct in* $\mathbb{E}$ *where* E *is a superclass and* $E_1, \ldots, E_n$ *are subclasses (partitions), the following must hold:*

  $E_i^I \subseteq E^I$ *for each* $i = 1, \ldots, n$ *and*

  $E_i^I \cap E_j^I = \emptyset$ *for each* $i \neq j$, *and*

  $E^I \subseteq E_1^I \cup \ldots \cup E_n^I$

- *for two connected levels* L_i, L_j *(each one being an entity or simple aggregation) in* $\mathbb{E}$ *there must be a (possibly partial) roll-up function* ρ_{L_i,L_j} *such that*

  $\rho_{L_i,L_j}(x) = y$ *for each* $x \in L_i^I$ *and* $y \in L_j^I$ $L_i, L_j \in \mathcal{E}$,

  We define reflexive transitive closure of roll-up function $\rho^*_{L_i,L_j}$ *(from* L_i *to any higher level* L_j *if there is a level* L_k *along the path between* L_i *and* L_j*)*
  inductively as follows:

  $\rho^*_{L_i,L_i} = \mathsf{id}$

  $\rho^*_{L_i,L_j} = \bigcup_k \rho_{L_i,L_k} \circ \rho^*_{L_k,L_j}$ *for each* k *such that* $L_i \ll L_k$

  where

  $$(\rho_{L_p,L_q} \cup \rho_{L_r,L_s})(x) = y$$

  $$\text{iff} \begin{cases} \rho_{L_p,L_q}(x) = \rho_{L_r,L_s}(x) = y, \text{ or} \\ \rho_{L_p,L_q}(x) = y \text{ and } \rho_{L_r,L_s}(x) = \perp, \text{ or} \\ \rho_{L_p,L_q}(x) = \perp \text{ and } \rho_{L_r,L_s}(x) = y \end{cases}$$

- *for each fact* $F \in \mathcal{F}$ *(being a weak entity) in* $\mathbb{E}$ *with* p *dimensions* $D_1, \ldots, D_p$ *(each one being an identifying relationship) and corresponding* p *levels* $L_1, \ldots, L_p$ *(each one being an entity or simple aggregation) in* $\mathbb{E}$ *for* $i = 1, \ldots, p$, $M_j \in \mathcal{A}$, $V_j \in \mathcal{V}$ *for* $j = 1, \ldots, m$,

  the following holds ($\mathcal{GMD}$ *cube conditions):*

  1. $\forall f.\ F(f) \rightarrow \exists l_1, \ldots, l_n.\ D_1(f) = l_1 \wedge L_1(l_1) \wedge \ldots \wedge D_n(f) = l_n \wedge L_n(l_n)$

  2. $\forall f, f', l_1, \ldots, l_n.\ F(f) \wedge F(f') \wedge D_1(f) = l_1 \wedge D_1(f') = l_1 \wedge \ldots \wedge D_n(f) = l_n \wedge D_n(f') = l_n \ \rightarrow \ f = f'$

- *for each aggregation* G *in* $\mathbb{E}$ *involving* n *dimensions* $D_1, \ldots, D_n$ *and* n *levels* $R_1, \ldots, R_n$ *(one per dimension) and a fact* F*:*

  $$G \doteq F\{D_1 \mid_{R_1}, \ldots, D_n \mid_{R_n}\}$$

  where $F \doteq E\{D_1 \mid_{L_1}, \ldots, D_p \mid_{L_p}\}$ *such that* $n \leq p$

  the following must hold:

  $$\forall g. \quad G^I(g) \quad \leftrightarrow$$
  $$g = \{\!|f \mid F^I(f) \wedge \bigwedge_{h=1,\ldots,p} (\rho^*_{L_h,R_h}(D_h^I(f)) = D_h^I(g))|\!\}$$

  for each $n \leq p$, $\{\!|\cdot|\!\}$ *denotes aggregation.*

Each aggregated cell is an aggregation of cells whose coordinates roll-up to the coordinates associated with an aggregated cell on which it is based on.

Thus, a particular EER diagram denotes a set of data warehouse states. According to $\mathcal{GMD}$, a particular EER schema is a set of *legal* data warehouse states, if they (data warehouse states) satisfy the cube (together with the aggregated cube) conditions imposed by the $\mathcal{GMD}$ schema, i.e., the set of all possible data warehouse states which conform to the constraints imposed by the $\mathcal{GMD}$ schema, conform to the diagram it self —i.e., they are legal data warehouse states. If a diagram is inconsistent, then no data warehouse may conform to it.

5 Related Work

Several proposals on a conceptual model exist in the data warehouse field. The only proposals by (Golfarelli et al. 1998, Sapia et al. 1998, Tryfona et al. 1999, Husemann et al. 2000, Zepeda & Celma 2006) address the conceptual model in a real fashion. The proposals by (Perez et al. 2005, Berenguer et al. 2005) address UML model (Perez et al. 2005) based on star schema, and propose the quality indicator metrics (Berenguer et al. 2005) for the conceptual model, although their work is based on UML modeling.

In (Golfarelli et al. 1998), a Dimensional Fact Model (DFM) is constructed from an operational ER schema based on requirement analysis. The construction methodology is well defined. It is relational and is based on star schema. DFM does not support generalisation/speciaisation hierarchies and many-to-many relationships. In the similar manner, Zepeda and Celema (Zepeda & Celma 2006) presented a Model Driven Architecture (MDA) for producing candidate multidimensional schemas from operational ER schema based on requirement analysis. Each of the candidate schema is based on star schema. However, a model supports generalisation hierarchies and many-to-many relationships. A mapping is presented for transformation of candidate (multidimensional) ER schema to cube, dimensions, levels, and measures. In both DFM and MDA, no aggregation is defined at conceptual schema level.

In (Kimball 1997, 1996), a multidimensional modeling manifesto using multidimensional view of enterprise data has been proposed; it is a relational implementation in the form of "star schema". This approach is not conceptual in the sense that it is not independent of the implementation.

A multidimensional conceptual model called MultiDimER model based on ER model has been proposed in (Malinowski & Zimanyi 2006). The model is based on star and snowflake schema. The features

such as generalisation/specialisation hierarchies, composite attributes, aggregations, etc have not been considered in this model. The model is well defined. It is based on the ER model and its logical representations. A conceptual model proposed by (Abello et al. 2006) is based on UML and its extensions, emphasizing on part-hole relationships for aggregation but does not support aggregations at the schema level.

None of these proposals addresses conceptual structure of aggregation. They only derive basic multidimensional schema from the given ER schema. Moreover, all these models need to specify design methodology such as information analysis, requirement analysis and specifications, etc (Golfarelli et al. 1998, Husemann et al. 2000) manually. The only proposal by (Franconi & Sattler 1999) for data warehouse conceptual model presents the structure of multidimensional aggregation; and it automates the construction of multidimensional conceptual schema from an ER diagram. The $\mathcal{CGMD}$ model is purely conceptual and addresses all the issues from data warehouse construction to aggregations and view management. The $\mathcal{CGMD}$ takes care of all constraints of the standard ER model in addition to multidimensional constraints. This shows $\mathcal{CGMD}$ is syntactically and semantically richer than the other models.

6 Evaluation of the $\mathcal{CGMD}$ model

In this section, we evaluate the $\mathcal{CGMD}$ data model according to certain criteria found for a multidimensional conceptual model in the literature. We also compare $\mathcal{CGMD}$ with other models. For evaluation, we consider some criteria listed in (Blaschka et al. 1998), nine requirements introduced in (Pedersen & Jensen 1999), and several requirements found in (Abello et al. 2001) for a data warehouse multidimensional model. We also consider some additional requirements which are also important for a data warehouse multidimensional conceptual model. All these requirements are randomly listed below.

1. *Implementation independent* (Blaschka et al. 1998):
2. *Explicit Separation of Structure and Contents* (Blaschka et al. 1998):
3. *Explicit hierarchies* (Blaschka et al. 1998, Pedersen & Jensen 1999): A model should support the explicit hierarchy in the dimension.
4. *Symmetric treatment for dimensions and measures* (Blaschka et al. 1998, Pedersen & Jensen 1999): A model should allow measures to be treated as dimensions and vice versa.
5. *Multiple hierarchies in dimension* (Pedersen & Jensen 1999):
6. *Dimension/level attributes* (Abello et al. 2001): A model should specify the attributes that do not define hierarchies.
7. *Support for aggregation* (Pedersen & Jensen 1999): A model should be able to provide meaningful aggregations.
8. *Complex Measures* (Blaschka et al. 1998): A model should support multiple and complex measures for the same fact (cube).
9. *Handling different levels of granularity* (Pedersen & Jensen 1999):
10. *Support for non-onto hierarchies* (Pedersen & Jensen 1999): A model should support non-onto (unbalanced) hierarchies, i.e., hierarchies with paths of different lengths.
11. *Support for non-strict hierarchies* (Pedersen & Jensen 1999): A model should support non-strict hierarchies.
12. *Support for many-to-many relationships* (Pedersen & Jensen 1999):
13. *Generalisation/specialisation hierarchies* (Abello et al. 2001): A model should support genenalisation/specialisation (is-a) relationships.
14. *Handling change over time* (Pedersen & Jensen 1999):
15. Handling uncertainty
16. *Multi-cube/fact schema* (Abello et al. 2001): A model should support multiple cubes/facts in schema.

    Since in $\mathcal{CGMD}$ one cube/fact is based on another, it ($\mathcal{CGMD}$) allows
17. *Summarisability:* A model should support summarisation (Lez & Shoshani 1997).
18. *User defined Aggregation functions* (Abello et al. 2001): A model should support user defined aggregation functions.
19. *Drill-across* (Abello et al. 2001): A model should allow to drill-across (sharing dimensions).
20. *Dimensionless aggregation*
21. *Measureless aggregation*
22. *Aggregation from aggregation (view over view)*

The $\mathcal{CGMD}$ model fulfills all the above requirements 1–22 except requirements 4, 12 and 14. Requirements 4 and 12 are partially supported, however, requirement 4 can be fully supported if the measure is used as coordinate. The requirement 14 is not supported as no syntactical provision is made for changing dimensions/levels in $\mathcal{CGMD}$. In addition, the $\mathcal{CGMD}$ model supports aggregations at any higher level (ignoring intermediate levels). This is one of the important characteristics of the $\mathcal{CGMD}$ model.

7 Comparison of $\mathcal{CGMD}$ with other models

In this section, we evaluate other models against the same requirements listed in Section 6, and compare them with the $\mathcal{CGMD}$ model which is already evaluated in Section 6.

We consider the models proposed in (Golfarelli et al. 1998, Sapia et al. 1998, Tryfona et al. 1999, Husemann et al. 2000) for comparison as they are conceptual models in true sense. However, models proposed in (Tsois et al. 2001, Abello et al. 2001, Pei 2003, Jensen et al. 2004), and (Trujillo et al. 2001) (an object oriented model— extension of (Trujillo et al. 2000)) are also taken into consideration for comparison because they are current state-of-the-art models, and are also conceptual in some way or other. Table in Figure 6 presents summary of comparison of these models. As before, if a model meets a particular requirement/feature/functionality *fully*, then it is denoted by "$\surd$". If a model supports the requirement partially then it denoted by "p", and if a model does not support it at all then it is denoted by "x".

Requirement 1 (Implementation independent) is partially supported by a model of (Abello et al. 2001), since it is a multi-star schema based on concepts of star model. Remaining models (Kimball 1996, Golfarelli et al. 1998, Jensen et al. 2004, Pei 2003), can not be considered implementation independent as they are either relational (Kimball 1996) or based on star schema ((Golfarelli et al. 1998)), a relational model or designed for a specific domain (Jensen et al. 2004, Trujillo et al. 2001, Pei 2003) implementation, for example, clinical domain (Jensen et al. 2004), and thus provide no support.

Requirement 2 (Explicit Separation of Structure and Contents), requirement 3 (Explicit hierarchies), requirement 5 (Multiple hierarchies) and requirement 8 (Complex measures) are met by all of the models (except star schema (Kimball 1996) for requirement 3), thus provide full support. Star schema does not support explicit hierarchy, and consequently does not support the requirement 3.

Requirement 6 (Dimension/level attributes). Only one model (Jensen et al. 2004) does specify the attributes that do not define hierarchies, hence provides no support. Remaining models specify the non-dimension attributes, thus provide full support.

Requirement 4 (Symmetric treatment for dimensions and measures), only one model (Jensen et al. 2004) captures this feature by means of derivation mechanism, thus providing full support. Some models (Abello et al. 2001, Tryfona et al. 1999) do not consider measure explicitly as dimensions but conditionally if the measure is used to identify a cell, thus providing partial support. Remaining models (Golfarelli et al. 1998, Sapia et al. 1998, Husemann et al. 2000, Tsois et al. 2001, Trujillo et al. 2001, Pei 2003), do not consider this feature in the framework and thus provide no support.

Requirement 7 (Support for correct aggregation) is met by a very few models (Abello et al. 2001, Jensen et al. 2004) either by derivation through some operations (Abello et al. 2001) or by restricting hierarchies to strict, covering and onto through some derivations so that data will not be double counted, thus provide full support. Only one model (Tsois et al. 2001) does not support this requirement because of including many-to-many relationships between facts and dimensions and among the hierarchies. Remaining models partially support this feature either by restricting the dimension/hierarchies and aggregation functions (Golfarelli et al. 1998) or hierarchies to strict, onto and covering, and restricting aggregation functions (Sapia et al. 1998, Husemann et al. 2000, Tsois et al. 2001, Trujillo et al. 2001, Pei 2003).

Requirement 9 (different levels of granularity) A very few models (Tsois et al. 2001, Abello et al. 2001) captures different levels of granularity (i.e., measures at different levels of granularity) either by aggregation (Tsois et al. 2001) or by specialising the cells depending on whether the measure is derived or not. Only one model (Jensen et al. 2004) captures this feature partially through some derivation. Rest of the models do not capture this feature, and hence provide no support.

For Requirement 10 (Support for non-onto hierarchies), only three models (Trujillo et al. 2001, Tsois et al. 2001, Jensen et al. 2004) fully support this features. Remaining models provide no support.

Requirement 11 (Support for non-strict hierarchies) is fully supported by only two models (Jensen et al. 2004, Tsois et al. 2001). Remaining models provide no support.

Requirement 12 (Support for many-to-many relationships). A model of (Tsois et al. 2001) supports many-to-many relationships between facts and dimensions but does not support many-to-many relationships between hierarchies. Only four models (Tryfona et al. 1999, Trujillo et al. 2001, Abello et al. 2001, Jensen et al. 2004) of the other models support both many-to-many relationships between facts and dimensions and between hierarchies. Rests do not support many-to-many relationships.

Requirement 13 (Generalisation/specialisation hierarchies). Support provided by (Sapia et al. 1998, Tryfona et al. 1999, Abello et al. 2001, Trujillo et al. 2001, Jensen et al. 2004) is considered partial, because generalisation/specialisation is considered in the hierarchy but is rather kept to distinguish the contents, thus providing partial support. Remaining models do not support this feature.

Requirement 14 (Change over time in data) and Requirement 15 (Uncertainty in data) are supported by a model (Jensen et al. 2004) by attaching a time tag attribute to dimension values for probabilistic measurements of occurrences of facts and dimension values. A model of (Abello et al. 2001) supports requirement 14 only by changing the schema with appropriate time tags, but does not support requirement 15. Remaining models do not support both features.

Requirement 16 (Multi-cube/fact schema). Only one model (Abello et al. 2001) allows multiple stars in a single schema. However, which dimensions/levels belong to which star schema are not clearly reflected in this model in any way, thus providing partial support. Remaining models do not meet this feature, thus provide no support.

Requirement 17 (Summarisability). The only models such as (Abello et al. 2001) and (Jensen et al. 2004) provide full support, either by applying some algebraic operations to make *part-whole* relationships between levels and then applying aggregation functions (Abello et al. 2001) or by computing weighting factor between facts and dimensions and makes full covering relationship between levels in the aggregation path (Jensen et al. 2004). Some models such as (Trujillo & Palomar 1998, Trujillo et al. 2000, 2001), (Golfarelli et al. 1998), and (Tryfona et al. 1999) specify possible functions that can be applied in order to support summarisation, but do not provide with a specific operation application, thus provide only partial support. Remaining models do not support summarisation.

Requirement 18 (user defined aggregation function). The only model of (Abello et al. 2001) supports this feature since it based on UML which allows user defined operations. Rest of the models provide no support.

Requirement 19 (Drill-across). The only model of (Abello et al. 2001) allows drill-across because of sharing multi-star dimensions, thus, providing full support. Some models such as Star (Kimball 1996) (constellation) and DFM (Golfarelli et al. 1998) share dimensions but limit drilling mechanism and hence provide partial support. Remaining models provides no support.

Requirement 20 (Dimensionless aggregation) and Requirement 22 (aggregation from aggregations) are not met by any of the other models. Requirement 21 (measureless aggregation) is partially supported by only two models star (Kimball 1996) and DFM (Golfarelli et al. 1998) by exploring the possibility of having factless (measureless) fact but they do not address or reflect aggregation in any way.

Model	Requiremets/Criteria/Features/Functionalities																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
star	x	p	x	p	√	p	√	x	x	x	x	x	p	x	x	p	p	x	p	x	p	x
DFM	√	√	√	x	√	√	p	√	x	p	x	x	x	x	x	p	p	x	p	x	p	x
ME/R	x	√	√	x	√	√	x	√	x	x	x	x	p	x	x	p	x	x	x	x	x	x
starER	x	√	√	p	√	√	p	√	x	x	x	√	p	x	x	x	p	x	x	x	x	x
DWPM	x	x	√	x	√	√	p	√	x	p	x	x	p	x	x	x	p	x	x	x	x	x
OOCM	x	√	√	x	√	√	p	√	x	√	x	√	p	x	p	p	x	x	x	x	x	x
MAC	x	√	√	x	√	√	x	√	√	√	√	√	x	x	x	x	x	x	x	x	x	x
YAM	√	√	√	p	√	√	√	√	√	x	x	√	p	p	x	√	√	√	√	x	x	x
GOLAP	x	√	√	x	√	√	p	√	x	p	x	√	x	x	x	x	p	x	x	x	x	x
EMDM	x	√	√	√	√	√	p	√	p	√	√	√	x	√	√	x	p	x	x	x	x	x
$\mathcal{CGMD}$	√	√	√	p	√	√	√	√	√	√	√	p	√	x	√	√	√	√	√	√	√	√

Figure 6: Comparison between $\mathcal{CGMD}$ and other Multidimensional Models

8 Conclusions and Future work

There are several proposals on multidimensional modeling and data warehouse design. So far, there is no consensus on modeling and design method yet (Rizzi & Abello 2006). The $\mathcal{CGMD}$ data model gives uniform way of modeling multidimensional concepts, data warehouse design and aggregations. Thus, generalising the design of data warehouse and providing the uniform way for view management. It is a framework where to translate and compare conceptual properties and expressive power of different expressivities of the models and related extensions. The $\mathcal{CGMD}$ model is to help cost effective design of the data warehouse, update propagation and view management of multidimensional data. Our future work is based translation of $\mathcal{CGMD}$ into Description Logic–a language for reasoning mechanism.

Acknowledgements

The major portion of this work was supported by School of Computer Science, University of Manchester, United Kingdom and Free University of Bolzano, Italy.

References

Abello, A., Samos, J. & Saltor, F. (2001), Yam (yet another multidimensional): An extension of uml, *in* 'Techniocal Report LSI-01-43-R of Department de Llenguates i Sistems Informatics'.

Abello, A., Samos, J. & Saltor, F. (2006), Yam2: A multidimensional conceptual model extending uml, *in* 'Information Systems, 31(6)', pp. 541–567.

Agrawal, R., Gupta, A. & Sarawagi, S. (1997), Modeling multidimensional databases, *in* 'Proc. of ICDE-97'.

Berenguer, G., Romero, R., Trujillo, J., Serrano, M. & Piattini, M. (2005), A set of quality indicator and thier corresponding metrics for conceptual models for data warehouses, *in* 'Proc. International Conference on Data Warehousing and Knowledge Discovery (DaWak'05)', pp. 95–104.

Blaschka, M., Sapia, C., Hofling, G. & Dinter, B. (1998), Finding your way through multidimensional data models, *in* 'Proc. 9th International Workshop on Database and Expert Systems Applications (DEXA'98)', pp. 198–203.

Borgida, A., Lenzerini, M. & Rosati, R. (2003), Description logics for databases., *in* F. Baader, D. Calvanese, D. McGuinness, D. Nardi & P. Patel-Schneider, eds, 'Description Logic Handbook', Cambridge University Press, chapter 16, pp. 462–484.

Calvanese, D., Lenzerini, M. & Nardi, D. (1998), Description logics for conceptual data modeling, *in* 'Chomicki, Jan and Saake, Gunter, editors 1998, Logics for Databases and Information Systems. Kluwer', pp. 229–264.

Elmasri, R. & Navathe, S., eds (2000), *Fundamentals of Database Systems, Third Edition*, Addison-Wesley.

Franconi, E. & Kamble, A. (2003), The $\mathcal{GMD}$ data model for multidimensional information: a brief introduction, *in* 'Proc. of 5th International Conference on Data Warehousing and Knowledge Discovery (DaWak-03)', pp. 55–65.

Franconi, E. & Kamble, A. (2004*a*), Data warehouse conceptual data model, *in* 'Proc. of 16th International International Conference on Scientific and Statistical Database Management (SSDBM)'.

Franconi, E. & Kamble, A. (2004*b*), The $\mathcal{GMD}$ data model and algebra for multidimensional information, *in* 'Proc. of 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), Riga, Latvia, June 7-11', pp. 446–462.

Franconi, E. & Sattler, U. (1999), A data warehouse conceptual data model for multidimensional aggregation, *in* 'Proc. of the Workshop on Design and Management of Data Warehouses (DMDW-99)', pp. 13–1–13–10.

Golfarelli, M., Maio, D. & Rizzi, S. (1998), 'The dimensional fact model: a conceptual model for data warehouses', *IJCIS* **7**(2-3), 215–247.

Husemann, B., Lechtenborger, J. & Vossen, G. (2000), Conceptual data warehouse modeling, *in* 'Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW'2000), Stockholm, Sweden, June 5-6', pp. 6–1–6–11.

Jensen, C., Kligys, A., Pedersen, T. & Timko, I. (2004), Multidimensional data modeling for location-based services, *in* 'The VLDB Journal Vol.13', pp. 1–21.

Kimball, R. (1996), *The Data Warehouse Toolkit*, John Wiley & Sons, USA.

Kimball, R. (1997), 'A dimensional modeling manifesto', *DBMS Magazine, August* **10**(9), 58–70.

Lez, H. J. & Shoshani, A. (1997), Summarizabilty in olap and statistical databases, *in* 'Proc. 9 th International Conference on Scientific and Statistical Database Management (SSDBM)'.

Malinowski, E. & Zimanyi, E. (2006), Hierrachies in a multidimensional model: From conceptual modeling to logical representation, *in* 'Data and Knowledge Engineering, In press'.

Moody, D. & Kortink, M. (2000), From enterpise models to dimensional models: A methodology for data warehouse and data mart design, *in* 'Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW'2000), Stockholm, Sweden, June 5-6', pp. 5–1–5–12.

Pedersen, T. & Jensen, C. (1999), Multidimensional data modeling for complex data, *in* 'Proc. of 15th IEEE Intenational Conference on Data Enginerring (ICDE-99)', pp. 336–345.

Pei, J. (2003), A general model for online analytical processing of complex data design and evolution, *in* 'Proc. of 22nd International Conference on Conceptual Modeling (ER2003)'.

Perez, J., Berlanga, R. & Pedersen, T. (2005), A relevence-extended multi-dimensional model for a data warehouse contextualized with documents, *in* 'Proc. 8th ACM International Workshop on Data Warehousing and OLAP (DOLAP'05)', pp. 19–28.

Phipps, C. & Davis, K. C. (2002), Automating data warehouse conceptual schema design and evolution, *in* 'Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW'2002)', pp. 23–32.

Rizzi, S. & Abello, A. (2006), Research in data warehouse modeling and design: Dead or alive?, *in* 'Proc. 9th ACM International Workshop on Data Warehousing and OLAP (DOLAP'06)', pp. 3–10.

Sapia, C., Blaschka, M., Hofling, G. & Dinter, B. (1998), Extending the er model for the multidimensional paradigm, *in* 'Proc. of ER Workshop', pp. 105–116.

Trujillo, J. & Palomar, M. (1998), An object oriented approach to multidimensional databases conceptual modelling, *in* 'Proc. of First International ACM Workshop Data Warehousing and OLAP', pp. 16–21.

Trujillo, J., Palomar, M. & Gomez, J. (2000), Applying object oriented conceptual modeling techniqyes to the design of multidimensional databases and olap applications, *in* 'Proc. of the First International Conference on Web-Age Information Management (WAIM'2000), Sanghai, China, June', pp. 83–94.

Trujillo, J., Palomar, M., Gomez, J. & Song, I. (2001), Designing data warehouses with oo conceptual models, *in* 'IEEE Computer Vol.34(12)', pp. 66–75.

Tryfona, N., Busborg, F. & Christiansen, J. (1999), starer: A conceptual model for data warehouse design, *in* 'Proc. of ACM Second International Workshop on Data Warehousing and OLAP (DOAP'99), Kansas City, Missouri, USA, November', pp. 3–8.

Tsois, A., Karayiannidis, N. & Sellis, T. (2001), MAC: Conceptual data modelling for OLAP, *in* 'Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW-2001)', pp. 5–1–5–13.

Zepeda, L. & Celma, M. (2006), A model driven approach for data warehouse conceptual design, *in* 'Proc. 7th IEEE International Baltic Conference on Databases and Information Systems (DBIS'06)', pp. 114–121.