A Data Warehouse Conceptual Data Model for Multidimensional Aggregation

Enrico Franconi
Dept. of Computer Science
Univ. of Manchester
Manchester M13 9PL, UK
franconi@cs.man.ac.uk

Ulrike Sattler
LuFG Theoretical Computer Science
RWTH Aachen
D-52074 Aachen, Germany
uli@cantor.Informatik.rwth-aachen.de

Abstract

This paper presents a proposal for a Data Warehouse Conceptual Data (CDWDM) Model which allows for the description of both the relevant aggregated entities of the domain—together with their properties and their relationships with other relevant entities—and the relevant dimensions involved in building the aggregated entities. The proposed CDWDM is able to capture the database schemata expressed in an extended version of the Entity-Relationship Data Model; it is able to introduce complex descriptions of the structure of aggregated entities and multiply hierarchically organised dimensions; it is based on Description Logics, a class of formalisms for which it is possible to study the expressivity in relation with decidability of reasoning problems and completeness of algorithms.

1 Introduction

Data Warehouse—and especially OLAP—applications ask for the vital extension of the expressive power and functionality of traditional conceptual modelling formalisms in order to cope with *aggregation*. Still, there have been few attempts [Catarci *et al.*, 1995, Cabibbo and Torlone, 1998] to provide such an extended modelling formalism, despite the fact that (1) experiences in the field of databases have proved that conceptual modelling is crucial for the design, evolution, and optimisation of a database, (2) a great variety of data warehouse system are on the market, most of them providing some implementation of multidimensional aggregation, and (3) query optimisation with aggregated queries [Nutt *et al.*, 1998, Cohen *et al.*, 1999] is even more crucial for data warehouses than it is for databases—which makes *semantic* query optimisation using a conceptual model even more important. As a consequence of the absence of a such an extended modelling formalism, a comparison of different systems or language extensions for query optimisation is difficult: a common framework in which to translate and compare these extensions is missing, new query optimisation techniques developed for extended schema and/or query languages cannot be compared appropriately.

In order to address these questions, a formal framework must be developed that encompasses the abstract principles of the data warehouse related extensions of traditional representation formalisms. In this paper, we present some preliminary outcome from the research done within the "Foundations of Data Warehouse Quality" (DWQ) long term research project, funded by the European Commission (n. 22469) under the ESPRIT Programme. With respect to the global picture, the role of our research within DWQ is to study a formal framework at the *conceptual level* (see Figure 1). The conceptual data model we are investigating should be able to abstract and describe the entities and relations which are relevant both in the whole enterprise, and in the user analysis of such information. In the following, we will refer to this formalism as the Data Warehouse Conceptual Data Model (DWCDM).

1.1 A Data Warehouse Conceptual Data Model

A DWCDM must provide means for the representation of a *multidimensional* conceptual view of data. More precisely, a DWCDM provides the language for defining multidimen-

The copyright of this paper belongs to the paper's authors. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage.

Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99)
Heidelberg, Germany, 14. - 15.6. 1999
(S. Gatziu, M. Jeusfeld, M. Staudt, Y. Vassiliou, eds.)
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-19/

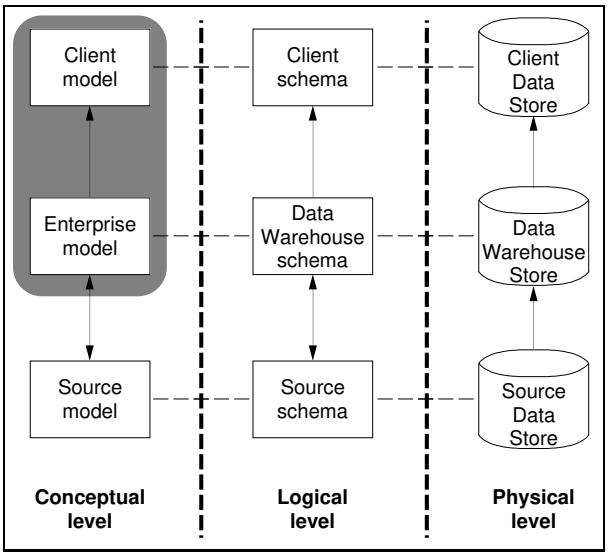

Figure 1: The role played by the Data Warehouse Conceptual Data Model with respect to the DWQ architecture.

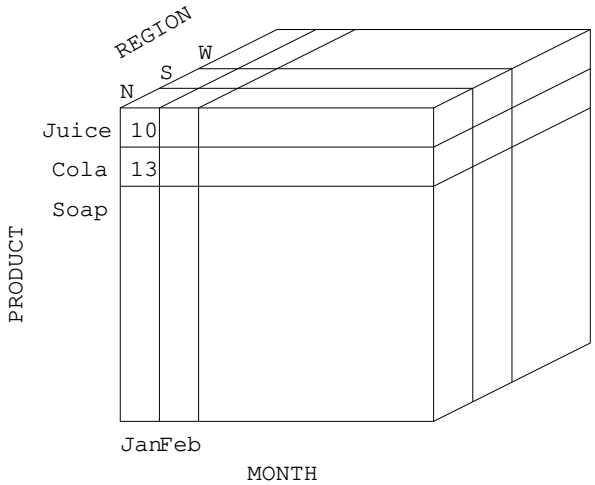

Figure 2: Sales volume as a function of product, time, location.

sional information within a conceptual model in the data warehouse global information base. As stated above, the model is of support for the conceptual design of a data warehouse, for query and view management, and for update propagation: it serves as a reference meta-model for deriving the inter-relations among entities, relations, aggregations, and for providing the integrity constraints necessary to reduce the design and maintenance costs of a data warehouse. Hence a DWCDM must be expressive enough to describe both the abstract business domain concerned with the specific application (*Enterprise model*)—just like a conceptual schema in the traditional database world—and the possible views of the enterprise information a specific user may want to analyse (*Client model*)—with particular emphasis on the aggregated views, which are peculiar to a data warehouse architecture (see Figure 1). A multidimensional modelling object in the logical perspective—e.g., a materialised view, a query, or a cube—should always be related with some (possibly aggregated) entity in the conceptual schema.

In the following, we will briefly introduce the ideas behind a multidimensional data model (see, e.g., [Agrawal *et al.*, 1995, Cabibbo and Torlone, 1998]) and compare it with a traditional relational data model. A more comprehensive introduction has been done in the forthcoming book "Fundamentals of Data Warehousing" [Baader *et al.*, 1999], Chapter 4 on *Multidimensional Aggregation*.

Relational database tables contain records (or rows). Each record consists of fields (or columns). In a normal relational database, a number of fields in each record (keys) may uniquely identify each record. In contrast, a multidimensional database contains n-dimensional arrays (sometimes called *hypercubes* or *cubes*), where each dimension has an associated hierarchy of levels of consolidated data. For instance, a spatial dimension might have a hierarchy with levels such as country, region, city, office.

Measures (which are also known as variables or metrics)—like Sales in the example, or budget, revenue, inventory, etc.—in a multidimensional array correspond to columns in a relational database table whose values functionally depend on the values of other columns. Values within a table column correspond to values for that measure in a multidimensional array: measures associate values with points in the multi-dimensional world. For example, the measure of the sales of the product Cola, in the northern region, in January, is 13,000. Thus, a dimension acts as an index for identifying values within a multidimensional array. If one member of the dimension is selected, then the remaining dimensions in which a range of members (or all members) are selected defines a sub-cube. If all but two dimensions have a single member selected, the remaining two dimensions define a spreadsheet (or a slice or a page). If all dimensions have a single member selected, then a single cell is defined. Dimensions offer a very concise, intuitive way of organising and selecting data for retrieval, exploration and analysis. Usual pre-defined or user-defined dimension levels (or Roll-Ups) for aggregating data in DW are: temporal (e.g., year vs. month), geographical/spatial (e.g., Rome vs. Italy), organisational (meaning the hierarchical breakdowns of your organisation, e.g., Institute vs. Department), and physical (e.g., Car vs. Engine).

A value in a single cell may represent an *aggregated* measure computed from more specific data at some lower level of the same dimensions. Aggregation involves computing *aggregation functions*—according to the attribute hierarchy within dimensions or to cross-dimensional formulas—for one or more dimensions. For example, the value 13,000 for the sales in January, may

Calls (av. duration)		Date (Day)								
		1/1/99	2/1/99	3/1/99	4/1/99	5/1/99	...	...	...	...
Source (Point Type)	Cell									
	Land Line			E_1						
	Direct Data									
	PABX									

Calls (av. duration)		Date (Week Day)						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Source (Customer Type)	Consumer					E_2		
	Business							

Figure 3: The cubes reporting the average duration of calls by dates in days and sources in point types, and by dates at the level of week days and sources at the level of customer types.

have been consolidated as the sum of the disaggregated values of the weekly (or day-by-day) sales. Another example introducing an aggregation grounded on a different dimension is the cost of a product—e.g., a car—as being the sum of the costs of all of its components.

In order to provide an adequate conceptualisation of multidimensional information, a DWCDM should provide the possibility of explicitly modelling the relevant *aggregations* and *dimensions*. According to a conservative point of view, a desirable DWCDM should extend some standard modelling formalism (such as Entity-Relationship) to allow for the description of both aggregated entities of the domain—together with their properties and their relationships with other relevant entities—and the dimensions involved. This document is about a proposal for a Data Warehouse Conceptual Data Model based on the Entity-Relationship model where aggregations and dimensions are first class citizens. The data model it is based on *Description Logics* (DL), which have been proved useful for a logical reconstruction of the most popular conceptual data modelling formalisms, including the (enhanced) ER model. Advantages of using Description Logics are their high expressivity combined with desirable computational properties—such as decidability, soundness and completeness of deduction procedures. The devised logic has a decidable reasoning problem, thus allowing for automated reasoning over the whole conceptual representation. The presented framework extends the ideas pursued in [Calvanese *et al.*, 1998b] regarding conceptual modelling using Description Logics as a data model, and the Information Integration framework presented in [Calvanese *et al.*, 1998a, Calvanese *et al.*, 1998c] based on an extended Description Logics data model for both the conceptual and the logical levels; our proposal is compatible with the DWCDM presented in [Calvanese *et al.*, 1998c].

The paper is organised as follows. Section 2 informally introduces an extended ER formalism which allows for the description of the explicit *structure* of multidimensional aggregations; the section briefly describes the semantics of the conceptual data model in terms of a logical representation of multidimensional databases, as proposed by [Cabibbo and Torlone, 1998]. Section 3 will propose a basic modelling language—based on Description Logics—which is expressive enough to capture the Entity-Relationship Data Model. The core part of the paper (Section 4) shows how it is possible to translate a schema expressed in the extended ER with aggregations in a suitable Description Logics theory, allowing for reasoning services such as satisfiability of a schema or the computation of a logically implied statement, such as an implicit taxonomic link between entities.

2 Modelling the Structure of Aggregation

We introduce in this section an extension of the Entity-Relationship Conceptual Data Model for representing the *structure* of aggregations. Thus, a conceptual schema will be able to describe abstract properties of multidimensional cubes, their interrelationships, and, most notably, their components. A Data Warehouse Conceptual Schema may contain detailed descriptions of the structure of aggregates,

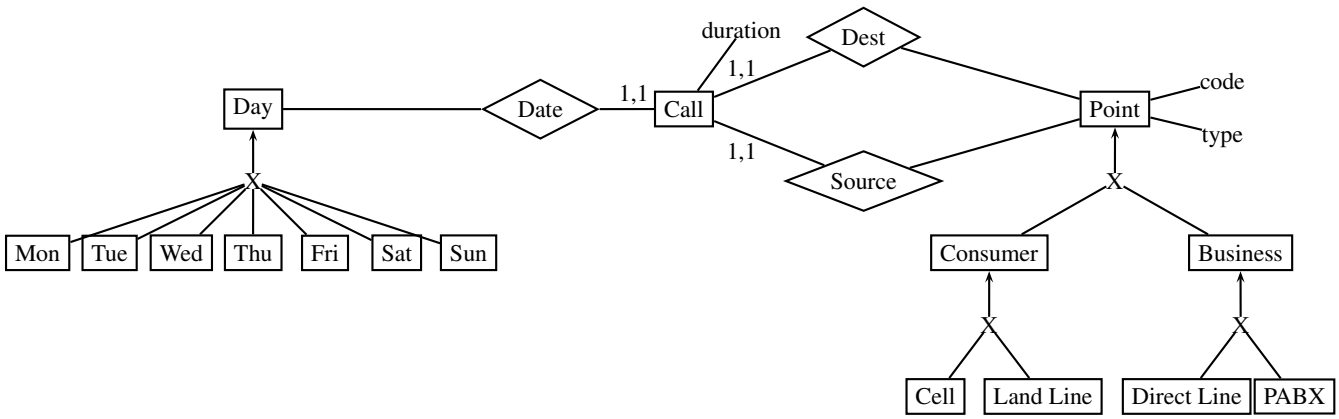

Figure 4: The Conceptual Data Warehouse Schema for the base data considered in Figure 3.

but it may not explicitly include aggregation functions.

Aggregations are first class citizens of the representation language: it is possible to describe the components of aggregations, and the relationships that the properties of the components may have with the properties of the aggregation itself; it is possible to build aggregations out of other aggregations, i.e., it is possible for an aggregation to be explicitly composed by other aggregations. This approach closely resembles the one pursued by [Catarci *et al.*, 1995, De Giacomo and Naggar, 1996], in the sense of proposing a conceptual data model in which aggregations are first-class entities intensionally described by means of their components.

As we have pointed out, the description of an aggregation is not going to include a specification of *how* values of its attributes are computed from attribute values of its components using aggregation functions such as min, average, or sum. While including such constructs in the conceptual model is obviously important, if we restrict our attention to data models which are computable (in a general sense), then we should be very conservative. The reason for this comes from an important result of the research within the DWQ project which identifies the borders for the possible extensions of a Data Warehouse Conceptual Data Model towards the explicit inclusion of aggregation functions [Baader and Sattler, 1998]. It has turned out that the *explicit* presence of aggregation functions, when viewed as a means to define new attribute values for aggregated entities, and built-in predicates in a concrete domain increases the expressive power of the basic conceptual model in such a way that all interesting inference problems may easily become undecidable. Moreover, this result is very tightly bounded: extending a very weak Conceptual Data Model allowing only basic constructs with a weak form of aggregation already leads to the undecidability of reasoning – i.e., no terminating procedure solving the reasoning problem may ever exist. On the other hand, recent research has shown that appropriate restrictions of the allowed aggregation functions yield decidability of these problems. These results concern (1) the use of aggregation functions in nested concepts, and (2) concrete domains like the integers, the non-negative integers, the rationals, and the reals.

2.1 An extended Entity-Relationship Model

As stated in [Agrawal *et al.*, 1995], a "good" data warehouse system should support user-definable *multiple* hierarchies along *arbitrary* dimensions. In Section 1.1 we have briefly defined a dimension as an index for identifying measures within a multidimensional data model. In the conceptual data model, "dimension" is a synonym for a domain of an attribute (or of attributes) that is structured by a hierarchy and/or an order. In order to support multiple hierarchies, the data model must provide means for defining and structuring these hierarchies, and for arbitrary aggregation along the hierarchies.

A conceptual data model where both multidimensional aggregations and multiply hierarchically organised dimensions can be abstracted and described can be used in query languages and for semantic optimization in multidimensional data bases. In fact, in the few attempts where a *cube algebra* introduces the notion of multiple dimensions and of levels within dimensions (e.g., [Cabibbo and Torlone, 1997, Vassiliadis, 1998]) the Data Warehouse Conceptual Schema can serve as a *reference meta-model* for deriving the inter-relations among levels and dimensions.

Let us now consider a concrete example related to the analysis of the average duration of telephone calls according to their dates and source types. The base data involved in the analysis is represented at the conceptual level in Figure 4. In order to perform the analysis, the two tables of Figure 3 are materialised by the OLAP tool. Each cell in the bi-dimensional cube on top denotes the aggregation composed by all the telephone calls issued at some date (expressed as a day of the year) and originated by a particular source (expressed as the type of the calling telephone);

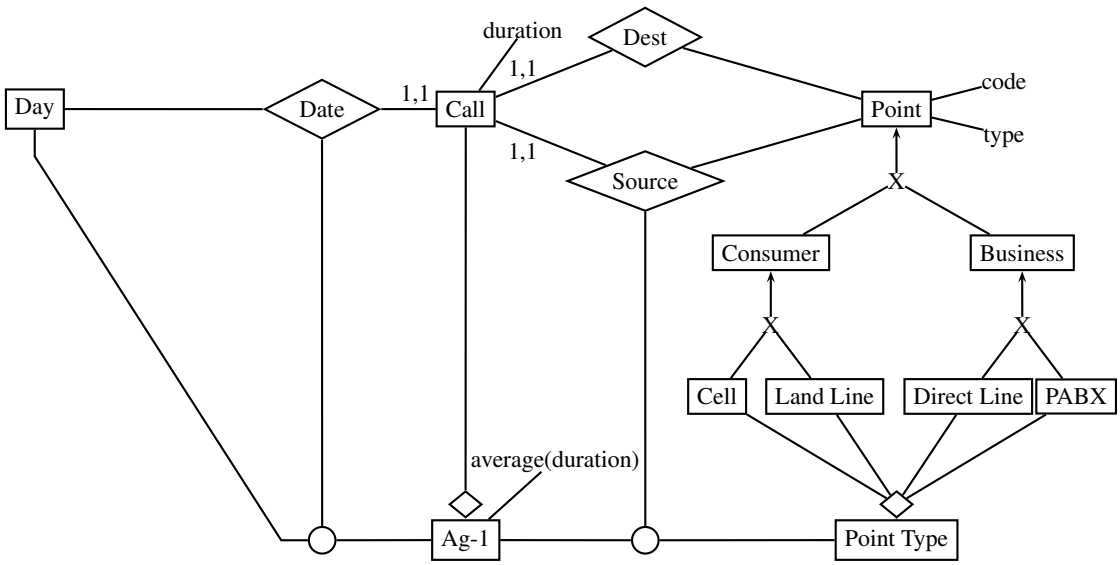

Figure 5: The Conceptual Data Warehouse Schema for the upper cube considered in Figure 3.

the date and the source are the *dimensions* of the cube, while the calls are the *target*. In particular, cell E_1 is the aggregation composed by all those calls issued on 3/1/99 and originating from a land line phone point. It is clear that E_1 may include more than one call, and it may itself have some properties which depend on all of its components. For example, E_1 may have the property `average(duration)` which denotes the average duration of all the calls issued on 3/1/99 and originating from a land line phone point. Of course, this property may be computed by an appropriate aggregation function from the property `duration` of the components.

An adequate basic conceptual schema for this simple multidimensional information base should include the base entities such as `Call`, `Day`, and `Phone Point` and relations such as `date` and `source`. Moreover, the schema should also include an additional *aggregated entity*, say `Ag-1`, namely the class denoting the aggregations of calls by date and source; such an aggregated entity can also have attributes such as `average(duration)`. We can also say that `Ag-1` aggregates telephone calls according to the (basic) level `Day` and the level `Point Type` of the dimensions `date` and `source`, respectively. The entity `Point Type` is itself an aggregation, aggregating all the specific telephone points according to their four basic types. It is clear that E_1 is one of the aggregations denoted by `Ag-1`.

Figure 5 presents the schema in a variant of the Entity-Relationship data model. The particular way of representing aggregated entities in the figure is inspired by [Catarci *et al.*, 1995, De Giacomo and Naggar, 1996].

If we also consider as part of the multidimensional information base the *aggregated view* represented by the second cube of Figure 3—denoting the aggregation composed by the telephone calls issued at some day of the week and originated from some source type of a different level as before (aggregated now according to consumer and business type points)—more conceptual entities come into play. Figure 6 presents the extensions required to the original schema.

Cell E_2 is the aggregation composed by all calls issued on Friday from a consumer type phone. Similar to E_1, E_2 may have the property `average(duration)` which *computes* the average duration of all those calls.

Thus, we need to add both a new aggregated entity and the definitions of the newly introduced levels for the dimensions `date` and `source`. The new aggregated entity, `Ag-2`, aggregates calls according to the level `Week day` and the level `Customer Type` of the dimensions `date` and `source` respectively. Then E_2 is one of the aggregations denoted by `Ag-2`. The level `Week Day` is obtained by aggregating days from the partitioning of the `Day` entity into seven sub-entities, namely the seven days of the week. The level `Customer Type` is obtained by aggregating phone points from the partitioning of the `Point` entity into the two sub-entities `Consumer` and `Business`. `Customer Type` is called simple aggregation, since there is no dimension involved in its definitions. `Customer Type` and `Week Day` are *levels* in the *multiply hierarchically organised* source and date dimensions.

We do not formally define in this paper the syntax of the extended ER model.

2.2 Semantics of the extended ER Model

The semantics of an ER schema is given in terms of legal multidimensional database states, i.e. multidimensional databases which conform to the constraints imposed by the schema. We consider as a starting point the ER semantics

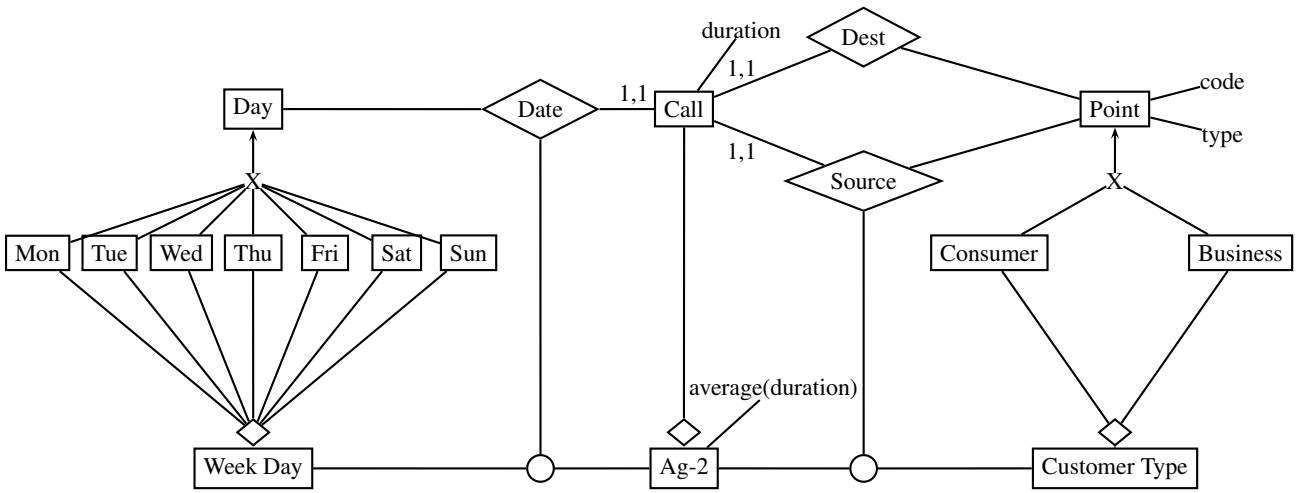

Figure 6: The Conceptual Data Warehouse Schema for the lower cube of Figure 3.

introduced in [Calvanese *et al.*, 1998b], recasted to cope with multidimensional information. For we have chosen the multidimensional logical data model $\mathcal{MD}$ introduced by [Cabibbo and Torlone, 1998]. $\mathcal{MD}$ is independent of any specific implementation of multidimensional databases (ROLAP or proprietary MOLAP), thus providing an abstract and general framework for the logical representation of multidimensional data. In [Cabibbo and Torlone, 1998] it is shown how a $\mathcal{MD}$ logical schema can be translated into a ROLAP logical representation in the form of a "star" schema, and into a general MOLAP logical representation in the form of sparse multidimensional arrays.

$\mathcal{MD}$ abstracts notions such as dimension hierarchies and levels, fact tables, cubes, and measures. As expected, dimensions are organised into hierarchies of levels, corresponding to the various granularity of the basic data. Within a dimension, levels are related through roll-up functions. The central element of a $\mathcal{MD}$ schema is the *f-table*, representing factual data. An *f-table* is the abstract logical representation of a multidimensional cube, and it is a function associating symbolic coordinates (one per involved dimension) to measures. According to the authors, a multidimensional database state is thus an *instance* of a $\mathcal{MD}$ logical schema: it is the description of the specific f-tables involved, in the form, for example, of tables describing the mapping from coordinates to measures.

Thus, a particular ER diagram denotes a set of multidimensional database states, i.e., the set of all possible multidimensional databases described as $\mathcal{MD}$ instances which conform to the diagram itself – i.e., they are legal states. If a diagram is inconsistent, then no multidimensional database may conform to it.

3 The basic Modelling Language

In this section we give a brief introduction to a basic Description Logic, which will serve as the basic representation language for our DWCDM proposal. With respect to the formal apparatus, we will strictly follow the concept language formalism introduced by [Schmidt-Schauß and Smolka, 1991] whose extensions have been summarised in [Donini *et al.*, 1996, Calvanese *et al.*, 1999].

The basic types of a concept language are *concepts*, *roles*, and *features*. A concept is a description gathering the common properties among a collection of individuals; from a logical point of view it is a unary predicate. Inter-relationships between these individuals are represented either by means of roles (which are interpreted as binary relations) or by means of features (which are interpreted as partial functions). Both roles and features can be used to individuals to certain properties. In the following, we will consider the Description Logic $\mathcal{ALCFI}$ [Horrocks and Sattler, 1999], extending $\mathcal{ALC}$ with features (i.e., functional roles), inverse roles, role composition, and role restrictions.

According to the syntax rules of Figure 7, $\mathcal{ALCFI}$ *concepts* (denoted by the letters C and D) are built out of *concept names* (denoted by the letter A), *roles* (denoted by the letter R, S), and *features* (denoted by the letters f, g); roles are built out of *role names* (denoted by the letter P) and features are built out of *feature names* (denoted by the letter p); it is worth noting that features are considered as special cases of roles.

Let us now consider the formal semantics of the $\mathcal{ALCFI}$. We define the *meaning* of concepts as sets of individuals—as for unary predicates—and the meaning of roles as sets of pairs of individuals—as for binary predicates. Formally, an *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a set $\Delta^{\mathcal{I}}$ of individuals (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$) mapping every concept C to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, every role R to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every feature f to a partial function

C, D	$\rightarrow$	$A \mid$	A	(concept name)
		$\top \mid$	`top`	(top)
		$\bot \mid$	`bottom`	(bottom)
		$\neg C \mid$	(`not` C)	(complement)
		$C \sqcap D \mid$	(`and` $C\ D\ \ldots$)	(conjunction)
		$C \sqcup D \mid$	(`or` $C\ D\ \ldots$)	(disjunction)
		$\forall R.C \mid$	(`all` $R\ C$)	(univ. quantifier)
		$\exists R.C \mid$	(`some` $R\ C$)	(exist. quantifier)
		$f\uparrow \mid$	(`undefined` f)	(undefinedness)
		$f : C$	(`in` $f\ C$)	(selection)
R, S	$\rightarrow$	$P \mid$	P	(role name)
		$f \mid$	f	(feature)
		$R^{-1} \mid$	(`inverse` R)	(inverse role)
		$R\|_C \mid$	(`restrict` $R\ C$)	(range restriction)
		$R \circ S$	(`compose` $R\ S\ \ldots$)	(role chain)
f, g	$\rightarrow$	$p \mid$	p	(feature name)
		$f \circ g$	(`compose` $f\ g\ \ldots$)	(feature chain)

Figure 7: Syntax rules for the $\mathcal{ALCFI}$ Description Logic.

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{i \in \Delta^{\mathcal{I}} \mid \forall j.(i,j) \in R^{\mathcal{I}} \Rightarrow j \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{i \in \Delta^{\mathcal{I}} \mid \exists j.(i,j) \in R^{\mathcal{I}} \wedge j \in C^{\mathcal{I}}\} \\
(f\uparrow)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus dom\, f^{\mathcal{I}} \\
(f : C)^{\mathcal{I}} &= \{i \in dom\, f^{\mathcal{I}} \mid f^{\mathcal{I}}(i) \in C^{\mathcal{I}}\} \\
\\
(R^{-1})^{\mathcal{I}} &= \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (j,i) \in R^{\mathcal{I}}\} \\
(R|_C)^{\mathcal{I}} &= R^{\mathcal{I}} \cap (\Delta^{\mathcal{I}} \times C^{\mathcal{I}}) \\
(R \circ S)^{\mathcal{I}} &= R^{\mathcal{I}} \circ S^{\mathcal{I}}
\end{aligned}
$$

Figure 8: The semantics of $\mathcal{ALCFI}$.

$f^{\mathcal{I}}$ from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}}$, such that the equations in Figure 8 are satisfied.

A *knowledge base*, in this context, is a finite set Σ of *terminological axioms*; it can also be called a *terminology* or TBox. For a concept name A, and (possibly complex) concepts C, D, terminological axioms are of the form $A \doteq C$ (concept definition), $A \sqsubseteq C$ (primitive concept definition), $C \sqsubseteq D$ (general inclusion statement). An interpretation $\mathcal{I}$ satisfies $C \sqsubseteq D$ if and only if the interpretation of C is included in the interpretation of D, i.e., $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. It is clear that the last kind of axiom is a generalisation of the first two: concept definitions of the type $A \doteq C$—where A is a concept name—can be reduced to the pair of axioms $(A \sqsubseteq C)$ and $(C \sqsubseteq A)$. Another class of terminological axioms—pertaining to roles R, S—are of the form $R \sqsubseteq S$. Again, an interpretation $\mathcal{I}$ satisfies $R \sqsubseteq S$ if and only if the interpretation of R—which is now a set of *pairs* of individuals—is included in the interpretation of S, i.e., $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$. A non-empty interpretation $\mathcal{I}$ is a *model* of a knowledge base Σ iff every terminological axiom of Σ is satisfied by $\mathcal{I}$. If Σ has a model, then it is *satisfiable*; thus, checking for KB satisfiability is deciding whether there is at least one model for the knowledge base. Σ *logically implies* an axiom α (written $\Sigma \models \alpha$) if α is satisfied by every model of Σ. We say that a concept C is *subsumed* by a concept D in a knowledge base Σ (written $\Sigma \models C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of Σ. A concept C is satisfiable, given a knowledge base Σ, if there is at least one model $\mathcal{I}$ of Σ such that $C^{\mathcal{I}} \neq \emptyset$, i.e. $\Sigma \not\models C \doteq \bot$. Concept subsumption can be reduced to concept satisfiability since C is subsumed by D in Σ if and only if $(C \sqcap \neg D)$ is unsatisfiable in Σ.

$\mathcal{ALCFI}$ was designed such that it is able to encode database schemas expressed in the most interesting Semantic Data Models and Object-Oriented Data Models [Horrocks and Sattler, 1999, Calvanese *et al.*, 1998b]. Recently, strictly more expressive conceptual data models based on DLs have been considered, most notably the $\mathcal{DLR}$ conceptual data modelling formalism. $\mathcal{DLR}$ was first introduced by [Calvanese *et al.*, 1998a] as a means for encoding conjunctive queries over expressive semantic data models for information systems such as extended Entity Relationship in the context of schema integration. We have chosen to limit the expressivity of the full $\mathcal{DLR}$ since we are looking for a language implementable with the current technology, but still capable to encode an interesting enhancement of the ER formalism. In particular, we have developed sophisticated reasoning algorithms for it [Horrocks and Sattler, 1999] and experimented them using the current academic implementations of expressive DLs, namely the systems FaCT [Horrocks, 1998] and iFaCT. It has been recently demonstrated [Horrocks and Patel-Schneider, 1999] that the logic we are considering here allows for the implementation of sound and complete reasoning algorithms that behave quite well both in realistic applications and systematic tests.

4 Encoding ER schemas with Aggregations

It is shown how a schema expressed in the conceptual data model informally introduced in the previous section can be expressed in an $\mathcal{ALCFI}$ knowledge base—whose models correspond with legal multidimensional database states of the ER diagram—allowing for reasoning services such as satisfiability of a schema or the computation of a logically implied statement.

In the following, we describe the translation between an ER diagram and an $\mathcal{ALCFI}$ knowledge base.

Definition 1 (Translation)
An ER schema $\mathcal{D}$ is translated into a corresponding knowledge base Σ where for each domain, entity, aggregation, or relationship symbol a concept name is introduced, and for

each attribute or ER-role symbol[1] symbol a feature name is introduced. The terminology Σ is defined to contain the following axioms:

- For each ISA link between two entities E, F (resp. two relationships R, S) in $\mathcal{D}$, Σ contains:
  $E \sqsubseteq F$ (resp. $R \sqsubseteq S$)

- For each PARTITION of an entity E into sub entities $F_1 \ldots F_n$ in $\mathcal{D}$, Σ contains:
  $E \sqsubseteq F_1 \sqcup \ldots \sqcup F_n$
  $F_i \sqsubseteq \neg F_j$ for all $i \neq j$
  $F_i \sqsubseteq E$ for all i

- For each attribute A in $\mathcal{D}$ with domain D of an entity E (resp. of a relationship R), Σ contains:
  $E \sqsubseteq A : D$ (resp. $R \sqsubseteq A : D$)

- For each relationship R in $\mathcal{D}$ relating n entities $E_1 \ldots E_n$ by means of the ER-roles $P^R_{E_1} \ldots P^R_{E_n}$, Σ contains:
  $R \sqsubseteq (P^R_{E_1} : E_1) \sqcap \ldots \sqcap (P^R_{E_n} : E_n)$

- For each minimum cardinality constraint $n = 1$ in an ER-role P^R_E in $\mathcal{D}$ relating a relationship R with and entity E (total or mandatory participation), Σ contains:
  $E \sqsubseteq \exists (P^R_E)^{-1}.R$

- For each aggregation Ag in $\mathcal{D}$ with targets $T_1 \ldots T_n$, Σ contains:
  $Ag \sqsubseteq (\exists \mathtt{target}.\top) \sqcap \forall \mathtt{target}.T_1 \sqcup \ldots \sqcup T_n$

- For each aggregation Ag in $\mathcal{D}$ involving a target T, n dimensions D_i (each one being a relationship in $\mathcal{D}$) and corresponding n levels L_i (each one being either an entity E_i or a simple aggregation Ag_i in $\mathcal{D}$), Σ contains:

$$
\begin{aligned}
Ag \sqsubseteq \forall \mathtt{target}.\ & \\
& ((\exists (P^{D_1}_T)^{-1}|_{D_1} \circ P^{D_1}_{L_1}.\top) \sqcap \\
& ((\forall (P^{D_1}_T)^{-1}|_{D_1} \circ P^{D_1}_{L_1}.L^1_1) \sqcup \cdots \sqcup \\
& (\forall (P^{D_1}_T)^{-1}|_{D_1} \circ P^{D_1}_{L_1}.L^{m_1}_1)) \sqcap \\
& \cdots \sqcap \\
& (\exists (P^{D_n}_T)^{-1}|_{D_n} \circ P^{D_n}_{L_n}.\top) \sqcap \\
& ((\forall (P^{D_n}_T)^{-1}|_{D_n} \circ P^{D_n}_{L_n}.L^1_n) \sqcup \cdots \sqcup \\
& (\forall (P^{D_n}_T)^{-1}|_{D_n} \circ P^{D_n}_{L_n}.L^{m_n}_n)))
\end{aligned}
$$

  where $L^j_i = E_i$ if the level i is described by an entity E_i; otherwise, if the level is described by a simple aggregation Ag_i, we use its targets $L^j_i = T^j_i$.

■

[1] ER-roles are the names given to the arguments of relationships; we assume that a unique name is given within a relationship to each ER-role, representing a specific participation of an entity in the relationship.

Extending the results of [Calvanese *et al.*, 1994] to the case of multidimensional databases, it can be proved that the translation is correct, in the sense that whenever a reasoning problem has a specific solution in the ER model, then the corresponding reasoning problem in the DL has a corresponding solution, and vice-versa. This is grounded on the fact that there is a precise correspondence between legal multidimensional databases of $\mathcal{D}$ and models of Σ. Thus, it is possible to exploit DL reasoning procedures for solving reasoning problems in the ER model. The reasoning problems we are mostly interested in are *consistency* of a ER schema—which is mapped to a satisfiability problem in the corresponding DL knowledge base—and *logical implication* within a ER schema—which is mapped to a logical implication problem in the corresponding DL knowledge base.

The proof is based by establishing the existence of two mappings from legal multidimensional database states of $\mathcal{D}$ to models of Σ and vice-versa. Informally speaking, the existence of the mappings ensures that, whenever an aggregation is satisfiable in Σ, then a non-empty mapping describing the corresponding f-table in $\mathcal{D}$ exists, and vice-versa. The same applies for level orderings and roll-up functions in $\mathcal{D}$. A more detailed sketch of the proof will be given in the full paper.

As a final remark, it should be noted that the high expressivity of DL constructs can capture an extended version of the basic ER model, which includes not only taxonomic relationships, but also arbitrary boolean constructs to represent so called generalized hierarchies with disjoint unions; entity definitions by means of either necessary or sufficient conditions or both, and integrity constraints expressed by means of generalised axioms [Calvanese *et al.*, 1998b].

Let us now consider the example introduced in Section 2. We start to (partially) formalise the schema of Figure 4, i.e., the base data. Please recall that every role name which appears in the translation of an ER schema in a Description Logic knowledge base—with the exception of the aggregation roles—is a functional role name.

$$
\begin{aligned}
&\mathtt{DATE} \sqsubseteq \mathtt{what} : \mathtt{Call} \sqcap \mathtt{when} : \mathtt{Day} \\
&\mathtt{SOURCE} \sqsubseteq \mathtt{what} : \mathtt{Call} \sqcap \mathtt{where} : \mathtt{Point} \\
&\mathtt{DEST} \sqsubseteq \mathtt{what} : \mathtt{Call} \sqcap \mathtt{where} : \mathtt{Point}
\end{aligned}
$$

$$
\begin{aligned}
&\mathtt{Point} \sqsubseteq \mathtt{Consumer} \sqcup \mathtt{Business} \\
&\mathtt{Consumer} \sqsubseteq \mathtt{Point} \sqcap \neg \mathtt{Business} \\
&\mathtt{Business} \sqsubseteq \mathtt{Point} \sqcap \neg \mathtt{Consumer}
\end{aligned}
$$

The partitioning of days into the seven day of the week is translated in a similar way.
The aggregated entity Customer Type is the simple aggregation of telephone points into two categories:

$$\mathtt{Customer\text{-}Type} \sqsubseteq$$

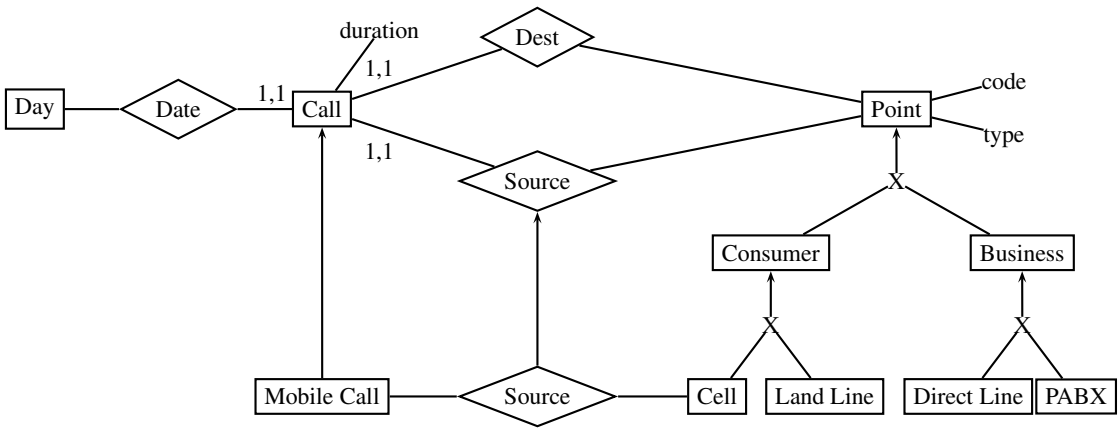

Figure 9: A Conceptual Data Warehouse Schema introducing the entity *Mobile Call*.

$(\exists \texttt{target}.\top) \sqcap$
$\forall \texttt{target}.(\texttt{Consumer} \sqcup \texttt{Business})$

The Week Day simple aggregation is obtained in a similar way.
The aggregated entity `Ag-2` is defined as being an aggregation composed by those calls issued in some day of the week and originated by either a consumer telephone point or a business telephone point:

$$
\begin{aligned}
\texttt{Ag-2} \sqsubseteq\ & (\exists \texttt{target}.\top) \sqcap \forall \texttt{target}.\texttt{Call} \\
\texttt{Ag-2} \sqsubseteq\ & \forall \texttt{target}.(\exists(\texttt{what}^{-1}|_{\texttt{SOURCE}} \circ \texttt{where}).\top \sqcap \\
& \quad (\forall(\texttt{what}^{-1}|_{\texttt{SOURCE}} \circ \texttt{where}).\texttt{Consumer} \sqcup \\
& \quad \forall(\texttt{what}^{-1}|_{\texttt{SOURCE}} \circ \texttt{where}).\texttt{Business}) \sqcap \\
& \quad \exists(\texttt{what}^{-1}|_{\texttt{DATE}} \circ \texttt{when}).\top \sqcap \\
& \quad (\forall(\texttt{what}^{-1}|_{\texttt{DATE}} \circ \texttt{when}).\texttt{Mon} \sqcup \\
& \quad \cdots \sqcup \\
& \quad \forall(\texttt{what}^{-1}|_{\texttt{DATE}} \circ \texttt{when}).\texttt{Sun}))
\end{aligned}
$$

Recall that `Ag-2` is the class of all aggregations such that each one of them aggregates calls issued at the same day of the week and originated from the same telephone point.

Each aggregation of calls belonging to the class denoted by `Ag-2` includes either only consumer originated calls or only business originated calls. In a similar way, each aggregation of `Ag-2` includes either only calls issued on Monday, or only calls issued on Tuesday, etc. Thus, aggregations denoted by `Ag-2` may be of fourteen possible types: Monday consumer, Monday business, Tuesday consumer, Tuesday business, etc.

As an example of reasoning, let us see a case with an inconsistent aggregation. If we introduce the entity Mobile Call as in Figure 9, it turns out that the aggregated entity having Mobile Call as target (instead of its super entity Call) and Business as level for the dimension Source is inconsistent, i.e., the materialised cube is necessarily empty. In fact, the translated theory in Description Logics turns out to be unsatisfiable, since mobile calls are originated only from cell points, which are disjoint from any kind of business phone point.

5 Conclusions

We have introduced a *Data Warehouse Conceptual Data Model*, extending the most interesting traditional Semantic Data Models and Object-Oriented Data Models, which allows the representation of a multidimensional conceptual view of data. We have seen how the proposed conceptual data model is able to introduce complex descriptions of the structure of aggregated entities and multiply hierarchically organised dimensions. In order to support multiple hierarchies, the data model provides means for defining and structuring these hierarchies, and for arbitrary aggregation along the hierarchies. Our future work will be devoted to a further development of the data model in order to explicitly consider temporal and spatial dimensions, and a study of the expressivity in relation with decidability and complexity of the *refinement* reasoning task.

References

[Agrawal *et al.*, 1995] Agrawal, R.; Gupta, A.; and Sarawagi, S. 1995. Modeling multidimensional databases. Technical report, IBM Almaden Research Center, San Jose, California. Proc. of ICDE'97.

[Baader and Sattler, 1998] Baader, Franz and Sattler, Ulrike 1998. Description logics with concrete domains and aggregation. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*. 336–340.

[Baader *et al.*, 1999] Baader, Franz; Franconi, Enrico; and Sattler, Ulrike 1999. *Multidimensional Data Models and Aggregation*. Springer-Verlag. chapter 4. Edited by M. Jarke, M. Lenzerini, Y. Vassilious and P. Vassiliadis.

[Cabibbo and Torlone, 1997] Cabibbo, Luca and Torlone, Riccardo 1997. Querying multidimensional databases. In *proc. Sixth Int. Workshop on Database Programming Languages (DBPL-97)*. 253–269.

[Cabibbo and Torlone, 1998] Cabibbo, Luca and Torlone, Riccardo 1998. A logical approach to multidimensional databases. In *Proc. of EDBT'98*.

[Calvanese *et al.*, 1994] Calvanese, Diego; Lenzerini, Maurizio; and Nardi, Daniele 1994. A unified framework for class-based representation formalisms. In *Proc. of KR-94*, Bonn D.

[Calvanese *et al.*, 1998a] Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Nardi, D.; and Rosati, R. 1998a. Description logic framework for information integration. In *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR-98)*. Morgan Kaufmann. 2–13.

[Calvanese *et al.*, 1998b] Calvanese, D.; Lenzerini, M.; and Nardi, D. 1998b. Description logics for conceptual data modeling. In Chomicki, Jan and Saake, Günter, editors 1998b, *Logics for Databases and Information Systems*. Kluwer.

[Calvanese *et al.*, 1998c] Calvanese, Diego; Giacomo, Giuseppe De; Lenzerini, Maurizio; Nardi, Daniele; and Rosati, Riccardo 1998c. Information integration: Conceptual modeling and reasoning support. In *Proc. of the 6th Int. Conf. on Cooperative Information Systems (CoopIS'98)*. 280–291.

[Calvanese *et al.*, 1999] Calvanese, Diego; De Giacomo, Giuseppe; Lenzerini, Maurizio; and Nardi, Daniele 1999. Reasoning in expressive description logics. In Robinson, Alan and Voronkov, Andrei, editors 1999, *Handbook of Automated Reasoning*. Elsevier Science Publishers, Amsterdam. To appear.

[Catarci *et al.*, 1995] Catarci, Tiziana; D'Angolini, Giovanna; and Lenzerini, Maurizio 1995. Conceptual language for statistical data modeling. *Data & Knowledge Engineering (DKE)* 17:93–125.

[Cohen *et al.*, 1999] Cohen, S.; Nutt, W.; and Serebrenik, A. 1999. Rewriting aggregate queries using views. In *Proc. of PODS'99*. To appear.

[De Giacomo and Naggar, 1996] De Giacomo, G. and Naggar, P. 1996. Conceptual data model with structured objects for statistical databases. In *Proceedings of the Eighth International Conference on Statistical Database Management Systems (SSDBM'96)*. IEEE Computer Society Press. 168–175.

[Donini *et al.*, 1996] Donini, F.; Lenzerini, M.; Nardi, D.; and Schaerf, A. 1996. Reasoning in description logics. In Brewka, G., editor 1996, *Principles of Knowledge Representation and Reasoning*. Studies in Logic, Language and Information, CLSI Publications. 193–238.

[Franconi and Sattler, 1999] Franconi, Enrico and Sattler, Ulrike 1999. A data warehouse conceptual data model for multidimensional aggregation: a preliminary report. *Journal of the Italian Association for Artificial Intelligence* AI*IA Notizie 9–21.

[Horrocks and Patel-Schneider, 1999] Horrocks, I. and Patel-Schneider, P. F. 1999. Optimising description logic subsumption. *Journal of Logic and Computation*. To appear.

[Horrocks and Sattler, 1999] Horrocks, Ian and Sattler, Ulrike 1999. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*. To appear.

[Horrocks, 1998] Horrocks, I. 1998. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6 th International Conference on Principles of Knowledge Representation and Reasoning*, Trento, Italy. 636–647.

[Nutt *et al.*, 1998] Nutt, Werner; Sagiv, Yehoshua; and Shurin, Sara 1998. Deciding equivalences among aggregate queries. In *Proc. of PODS'98*. 214–223.

[Schmidt-Schauß and Smolka, 1991] Schmidt-Schauß, M. and Smolka, G. 1991. Attributive concept descriptions with complements. *Artificial Intelligence* 48(1):1–26.

[Vassiliadis, 1998] Vassiliadis, P. 1998. Modeling multidimensional databases, cubes and cube operations. In *Proc. of the 10th SSDBM Conference*, Capri, Italy.