Hetero-Homogeneous Hierarchies in Data Warehouses

Bernd Neumayr[1] **Michael Schrefl**[1] **Bernhard Thalheim**[2]

[1] Department of Business Informatics - Data & Knowledge Engineering
Johannes Kepler University Linz, Austria
E-Mail: {neumayr,schrefl}@dke.uni-linz.ac.at

[2] Christian-Albrechts-University Kiel, Institute of Computer Science, Kiel, Germany
Email: thalheim@is.informatik.uni-kiel.de

Abstract

Data Warehouses facilitate multi-dimensional analysis of data from various data sources. While the original data sources are often heterogeneous, current modeling and implementation techniques discard and, thus, cannot exploit these heterogeneities.

In this paper we introduce *Hetero-Homogeneous Hierarchies* to model dimension hierarchies and cubes with inherent heterogeneities. Hetero-homogeneous hierarchies are hierarchies that are heterogeneous in regard to the schema of sub-hierarchies and homogeneous in regard to a minimal common schema shared by all sub-hierarchies.

Sub-dimension-hierarchies can be specialized to contain additional levels and additional non-dimensional attributes. Sub-cubes can be specialized towards additional measures, more fine-grained facts, and differing units of measure. We show how scale differences and conflicts due to multi-dimensional inheritance can be avoided and solved. We provide a formal definition of our approach together with a query/cube algebra.

Keywords: Multidimensional conceptual modeling, abstraction, specialization; Heterogeneous information; OLAP

1 Introduction

Data Warehouses facilitate multi-dimensional analysis of data integrated from various data sources. Available and interesting data is often heterogeneous concerning available measures, granularity of measures, units of measures, applicable rollup-levels, and interesting secondary information (non-dimensional attributes). However, to ease querying and storing multi-dimensional data, current modeling and implementation techniques force to fully homogenize available data according to a global multi-dimensional schema.

Our approach is summarized by the oxymoron term *hetero-homogeneous hierarchies.* A hetero-homogeneous hierarchy is a hierarchy with a single root node that is (1) homogeneous in regard to a minimal common schema shared by all sub-hierarchies, where a sub-hierarchy is a hierarchy rooted in a child of the root node, (2) heterogeneous in regard to the specialized schemas of sub-hierarchies.

We discuss our approach by a running example, starting with a homogeneous schema that can be

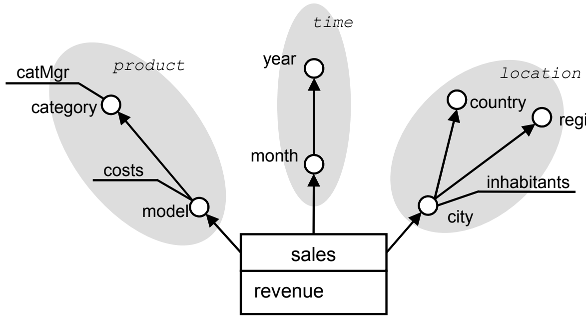

Figure 1: Homogeneous cube modeled with the Dimensional Fact Model

modeled using the Dimensional Fact Model (Golfarelli et al., 1998) (see Fig. 1). Consider a homogeneous *sales*-cube with dimensions *product*, *time*, and *location*. Dimension *product* defines dimension level *category* with non-dimensional attribute *catMgr* and dimension level *model* with non-dimensional attribute *costs*. The level-hierarchy of *product* defines *category* to be above *model*. Dimension *time* defines levels *year* above level *month*. Dimension *location* defines level *country* and level *region*, and level *city* with non-dimensional attribute *inhabitants*. Level *city* is below levels *country* and *region*, which are in no order to each other. The *sales*-cube defines a measure *revenue*.

Dimension hierarchies can be hetero-homogeneous with regard to

- *non-dimensional attributes*, e.g., in sub-hierarchy *Car* of dimension *product*, dimension instances at level model have an additional non-dimensional attribute *maxSpeed*.
- *additional levels*, e.g., in sub-hierarchy *Switzerland* of dimension *location* there is an additional level *kanton* between *city* and *country* and an additional level *store* below city.

Cubes can be hetero-homogeneous in that

- sub-cubes, such as *car sales in Switzerland 2009* may have *additional measures*, e.g., *quantity sold*,
- different sub-cubes may give *different units* for the same measure, e.g., values of measure *revenue* are provided in *swiss francs*
- base facts for various measures are provided at *mixed granularities*, e.g., base facts for *cheapestOffer* are provided at level *category*, *year*, *country* while base facts for measure *revenue* are provided at level *model*, *month*, *year*.

- different sub-cubes may provide base facts for the same measure at *different granularities*, e.g., measure *revenue* originally defined for level *model*, *month*, and *city* is now available more detailed at level *model*, *month*, and *store*.

To represent such kind of situations one needs a design approach to represent hetero-homogeneous hierarchies in dimensions and cubes. Such an approach should allow for an instance-based specialization of dimensions and cubes.

In previous work (Neumayr et al., 2009) we introduced multilevel-objects (m-objects) and multilevel-relationships (m-relationships) to represent objects and relationships at multiple levels of abstraction. In this paper, we show how hetero-homogeneous hierarchies in data warehouses can be modeled by a revised and extended form of m-objects and m-relationships.

Hetero-homogeneous dimension hierarchies are modeled as concretization hierarchies of m-objects. Thereby an m-object encapsulates and arranges dimension levels in a partial order from abstract to concrete. Thereby, it describes itself and the common properties of the objects at each level of the dimension hierarchy beneath itself. An m-object that concretizes another m-object inherits dimension levels and non-dimensional attributes of the parent. It may also introduce additional levels and additional non-dimensional attributes. For modeling dimension hierarchies, we extend the original definitions of m-objects to partially ordered level hierarchies and consistency criteria that avoid conflicts due to multiple concretization.

Cube schemas as well as facts are modeled as m-relationships. M-relationships are analogous to m-objects in that they describe relationships between m-objects at multiple levels of abstraction. For modeling cube schemata and facts, we extend m-relationships from binary m-relationships (Neumayr et al., 2009) to n-ary m-relationships that may define measures and assert measure values. We define consistency criteria that avoid conflicts due to multiple inheritance and avoid overlapping primary fact instances.

Most current approaches to data warehousing are centered around the notion of a *cube*, our conceptual approach to modeling and querying data warehouses is centered around multi-level cubes (m-cubes). An *m-cube* represents a cube of cubes, given by the cartesian product of dimension levels and, on a more fine-grained level, a set of coordinates, given by the cartesian product of dimension instances.

We also introduce an m-cube-algebra with closed m-cube operations *dice*, *slice*, *import-union*, and *projection*; together with *fact-* and *cube-extraction* operations. Other common data warehouse operations like roll-up, drill-down, drill-across are subsumed by these operations. To cope with heterogeneous measure units we also support unit conversion. In order to exploit heterogeneities in m-cubes queries are typically double-staged: after selecting a sub-m-cube, using dice, the query can make use of additional schema information like additional measures, refined granularity, additional non-dimensional attributes, and additional cube levels.

The paper is structured as follows: in Sec. 2 and Sec. 3 we show how to model hetero-homogeneous dimension hierarchies and hierarchies of m-cubes, respectively, and provide structural definitions and consistency criteria. In Sec. 4 we show how to query m-cubes and introduce an m-cube-algebra. In Sec. 5 we briefly survey related work. In Sec. 6, which concludes the paper, we give an outlook on future work.

2 Hetero-Homogeneous Dimension Hierarchies

In this section we first revisit and extend m-objects (Neumayr et al., 2009) and, then, we show how to model hetero-homogeneous dimensions with them.

2.1 M-Objects revisited

An m-object, as originally introduced, encapsulates and arranges abstraction levels in a linear order from the most abstract to the most concrete one. Thereby, it describes itself and the common properties of the objects at each level of the concretization hierarchy beneath itself. An m-object specifies concrete values for the properties of its top-level. This top-level describes the m-object itself. All other levels describe common properties of m-objects beneath itself.

We now give revised definitions that support m-objects with a partial (non-linear) order of levels.

Definition 1 (M-Object). *An m-object o is described by a 6-tuple $(L_o, A_o, P_o, l_o, d_o, v_o)$ where $L_o \subseteq L_D$ is a set of levels from a universe of levels and $A_o \subseteq A_D$ is a set of attributes from a universe of attributes. The levels L_o are organized in a partial order, as defined by parent relation $P_o \subseteq L_o \times L_D$, which associates with each level its parent levels. Each attribute is associated with one level, defined by function $l_o : A_o \to L_o$, and has a domain, defined by function $d_o : A_o \to datatypes$. Optionally, an attribute has a value from its domain, defined by partial function $v_o : A_o \to V$, where V is a universe of data values, and $v_o(a) \in d_o(a)$ iff $v_o(a)$ is defined.*

An m-object has a single top-level, $\hat{l}_o := l \in L_o : \nexists l' \in L_o : (l, l') \in P_o$.

We say o is at level l, if l is its top-level. We further say level l' is a child of level l iff $(l', l) \in P_o$, and l' is a descendant of, or below, l iff $(l', l) \in P_o^+$, where P_o^+ is the transitive closure of P_o, and l' is a descendant of or the same as l iff $(l', l) \in P_o^*$, where P_o^* is the transitive-reflexive closure of P_o.

M-objects, levels, and attributes have names, defined by function $name : O \cup L \cup A \to names$, where *names* is the universe of names. Names of m-objects, attributes, and levels are unique within one dimension.

Example 1 (M-Object Car). Product category *car* (see Fig. 2) has three levels *category*, *brand*, and *model* and defines a value for attribute *catMgr*.

An m-object can *concretize* another m-object, which is referred to as its parent, by introducing new levels, introducing new attributes, and providing values for attributes. The concretizes-relationship comprises classification, generalization and aggregation. A concretization relationship between two m-objects does not reflect that one m-object is at the same time an *instance of*, *component of*, and *subclass of* another m-object as a whole. Rather, a concretization relationship has to be interpreted in a multi-faceted way. This is exemplified by the following example.

Example 2 (Concretization). M-object *Car* concretizes *Product*. The concretization relationship is to be interpreted in a multi-faceted way: m-object *Car* is instance of level *category* of m-object *Product* because level *category*, which is the first non-top-level of m-object *Product*, is its top-level. It also specifies a value for its attribute *catMgr*. M-object *Car* specializes m-object *Product* by introducing a new level *brand* and adding attribute *maxSpeed* to level *model*. The level *model* of m-object *Car* is regarded as a subclass of level *model* of m-object *Product*.

A child m-object o' chooses its single top-level from the common second-top-levels of its parent m-objects. It 'inherits' from each parent m-object o all levels below its own top-level, together with the relative order of these common levels. It also 'inherits' attributes associated with common levels, together with the properties of these attributes, as defined by functions l_o, d_o, and v_o. In the case of multiple concretization the top-level of the child m-object must be a common second-top level of the parent m-objects.

For simplicity, we do not define this inheritance mechanism and assume that each m-object is fully described. We summarize the consistency criteria in the following definition.

Definition 2 (Consistent Concretization). *An m-object o' is a* consistent concretization *of another m-object o iff*

1. *The top-level of o' is a second-top-level in o: $(\hat{l}_{o'}, \hat{l}_o) \in P_o$*
2. *Each level of o, from $\hat{l}_{o'}$ downwards, is also a level of o': $l \in L_o : (l, \hat{l}_{o'}) \in P_o^* \Rightarrow l \in L_{o'}$ (*level containment*)*
3. *All attributes of o, associated with a level that is shared by o and o', also exist in o', $\{a \in A_o \mid l_o(a) \in L_{o'}\} \subseteq A_{o'}$ (*attribute containment*)*
4. *The relative order of common levels of o and o' is the same: $l, l' \in (L_{o'} \cap L_o) : (l, l') \in P_{o'}^+ \Leftrightarrow (l, l') \in P_o^+$ (*level order compatibility*)*
5. *Levels newly introduced in o' have parents only within o': $\forall (l, l') \in P_{o'} : l \in (L_{o'} \setminus L_o) \Rightarrow l' \in L_{o'}$ (*locality of level order*).*
6. *Common attributes are associated with the same level, have the same domain, and the same value, if defined: For $a \in (A_{o'} \cap A_o)$:*
   - *(a) $l_o(a) = l_{o'}(a)$ (*stability of attribute levels*)*
   - *(b) $d_o(a) = d_{o'}(a)$ (*stability of attribute domains*)*
   - *(c) $v_o(a)$ is defined $\Rightarrow v_o(a) = v_{o'}(a)$ (*compatibility of attribute values*)*

2.2 Modeling Hetero-Homogeneous Dimension Hierarchies with M-Objects

We now describe how a homogeneous dimension hierarchy can be modeled by m-objects: (1) The dimension is represented by a hierarchy of m-objects. (2) Each dimension level corresponds to a level of the root m-object. (3) Each level schema is represented by the attributes associated with that level of the root m-object. (4) A dimension instance of some dimension level is represented by an m-object, whose top-level is the dimension-level. (5) Attribute values associated with the top-level of an m-object describe the dimension instance that the m-object represents.

Example 3 (Homogeneous Dimension Hierarchies). Consider Fig. 4 ignoring all relationship symbols. M-objects *Product*, *Time*, and *Location* represent the dimensions of the Dimensional Fact Model depicted in Fig. 1. The m-object beneath the gray line depict dimension instances.

Additional non-dimensional attributes can be introduced at various levels for the successors of some dimension instance as follows: The m-object representing this dimension instance is extended by attribute definitions at that level; the m-object now

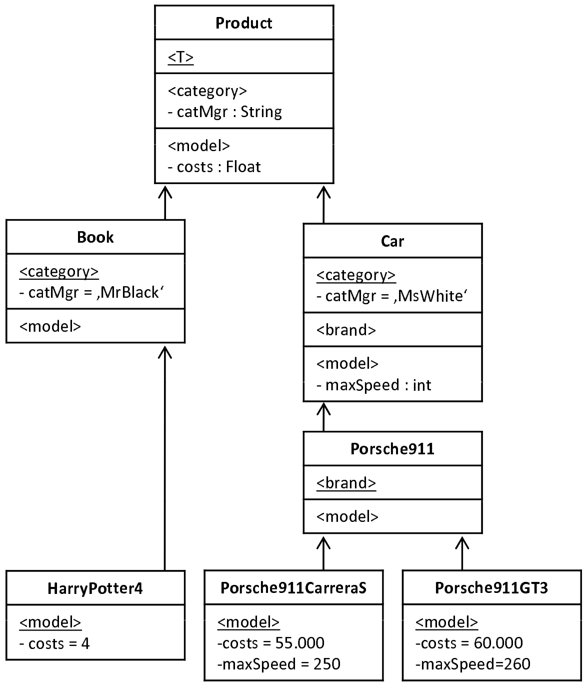

Figure 2: Hierarchy of m-objects representing hetero-homogeneous dimension hierarchy *product*. Attributes are only shown at m-objects where they are introduced or instantiated

serves also as dimension schema for the sub-hierarchy rooted at this dimension instance.

Additional levels can be introduced for the successors of some dimension instance as follows: The m-object representing this dimension instance is extended with additional levels and now serves also as dimension schema for the sub-hierarchy rooted in this dimension instance.

Example 4 (Hetero-homogeneous dimension hierarchy). In the dimension hierarchy *product* (see Fig. 2), m-object *car* introduces additional attribute *maxSpeed* at level *model* and additional level *brand*.

A data warehouse comprises multiple dimensions. Each dimension D organizes a set of m-objects $O_D \subseteq O$ in a hierarchy H_D, with levels L_D, taken from a universe of levels L, and describes m-objects using attributes A_D, taken from a universe of attributes A. Each m-object, but the root-m-object, has one or more parent-m-objects as defined by acyclic relation $H_D : O_D \times O_D$. Let $o, o' \in O_D$, then o' is said to be a *direct concretization of o* or *o' concretizes o*, iff $(o', o) \in H_D$, to be an *indirect concretization of o* iff $(o', o) \in H_D^+$, to be equal to or an indirect concretization of o iff $(o', o) \in H_D^*$. H_D^+ and H_D^* denote the transitive and transitive-reflexive closure, resp., of H_D.

In case of multiple concretization, stemming from level hierarchies that are not in a total but only in a partial order (see Fig. 3), we avoid conflicts due to 'multiple inheritance' by ensuring that each attribute and each level is inducted at exactly one m-object. We only consider dimensions with such concretizations of m-objects to be consistent.

Definition 3 (Consistent Dimension). *A dimension $D = (O_D, A_D, L_D, H_D)$ is* consistent*, iff*

1. *Each $o \in O_D$ is an m-object according to Definition 1.*
2. *For each pair of m-objects $(o', o) \in H_D$, o' is a consistent concretization of o according to Definition 2.*

3. *Each attribute and level is introduced at only one m-object:*
   (a) $a \in (A_o \cap A_{o'}) : \exists \bar{o} \in O : (o, \bar{o}) \in H_D^* \wedge (o', \bar{o}) \in H_D^* \wedge a \in A_{\bar{o}}$ *(*unique induction rule for attributes*)*
   (b) $l \in (L_o \cap L_{o'}) : \exists \bar{o} \in O : (o, \bar{o}) \in H_D^* \wedge (o', \bar{o}) \in H_D^* \wedge l \in L_{\bar{o}}$ *(*unique induction rule for levels*)*
4. *If an m-object o' with top-level l is a direct or indirect concretization of m-object o where $(l, l') \in P_o$ then o' must concretize an m-object $\hat{o}$ with top-level l'.*
5. *An m-object o may not directly or indirectly concretize two m-objects o', o'' that are at the same level, i.e., $(o, o') \in H^* \wedge (o, o'') \in H^* \Rightarrow \hat{l}_{o'} \neq \hat{l}_{o''}$ (*unique level predecessor*)*

Levels in a dimension, L_D, are implicitly partially ordered. This follows from the unique induction rule for levels and level order compatibility. We say, $l' \in L_D$ is a descendant of $l \in L_D$, written as $l' \prec l$, if there is an m-object $o \in O_D$ in which l' is a descendant of l. We write $l' \preceq l$ to denote that l' is either descendant of or equal to l. Also note that $\prec$ and $\preceq$ are transitive, i.e.: $\forall l', l \in L_D : (\exists o \in O_D : (l', l) \in P_o^*) \vee (\exists l'' \in L_D : l' \preceq l'' \wedge l'' \preceq l) \Rightarrow l' \preceq l$.

Example 5 (Consistent Hetero-homogeneous dimension hierarchy)**.** Consider dimension hierarchy *location* in Fig. 3, m-object Lausanne is an indirect concretization of m-object *location* via kanton *Vaud* and country *Switzerland*. As level *region* is also a parent level of level *city* in m-object *location*, Lausanne must also concretize an m-object at level *region*. This is with m-objects *Alps* the case.

3 Hetero-Homogeneous Cubes

In this section we revisit and extend definitions of m-relationships (Neumayr et al., 2009) and, then, we show how to model hetero-homogeneous cubes with them.

3.1 M-Relationships revisited

M-relationships as introduced in (Neumayr et al., 2009) are analogous to m-objects in that they describe relationships between m-objects at multiple levels of abstraction. They have the following features: (1) M-relationships at different abstraction levels can be arranged in concretization hierarchies, similar to m-objects. (2) An m-relationship represents different abstraction levels of a relationship, namely one relationship occurrence and multiple relationship classes. Such a relationship class collects all descending m-relationships that connect m-objects at the respective levels. (3) An m-relationship implies extensional constraints for its concretizations at multiple levels. (4) M-relationships can cope with heterogenous hierarchies and (5) m-relationships can be exploited for querying and navigating.

While our original approach considered only binary m-relationships without relationship attributes, the revised definition below covers for n-ary m-relationships that are described by attributes. Taking into account the data warehouse context the attributes are measures, have an associated aggregation function, and a connection level indicating at which detail measure values are provided.

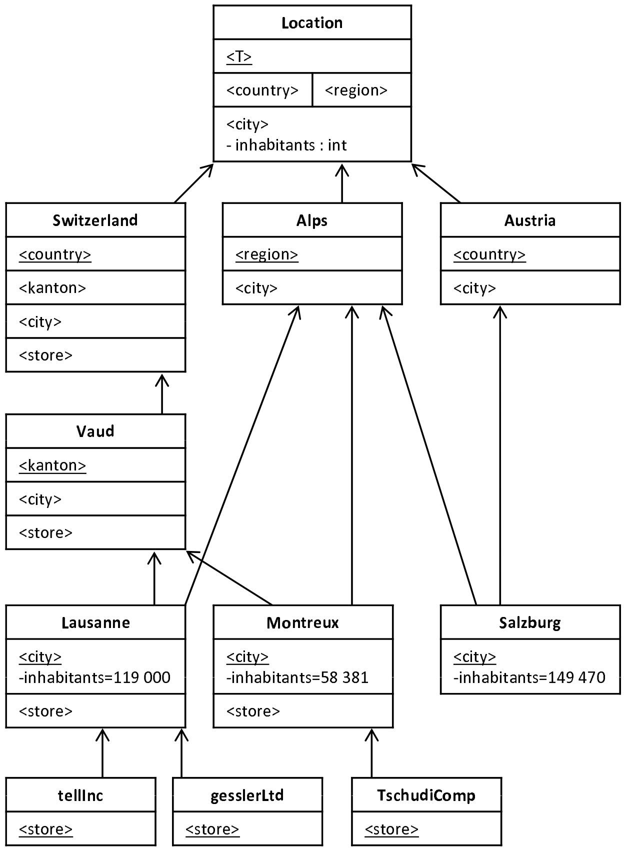

Figure 3: Hetero-homogeneous dimension hierarchy *location* with multiple concretization

Definition 4 (M-Relationship)**.** *An m-relationship $r = (o_1, ..., o_n; M, b, u, f, v)$ between m-objects $o_1, ..., o_n$, its coordinate (denoted also by* coord(r)*), is described by a set of measures M. Its top-connection-level $\hat{l}_r$ is implicitly given by the top-levels of the referenced m-objects, i.e., $\hat{l}_r := (\hat{l}_{o_1}, ..., \hat{l}_{o_n})$. Each measure $m \in M$ is described by*

1. *a connection-level, as defined by total function $b : M \rightarrow (L_{o_1} \times ... \times L_{o_n})$*
2. *a unit of measure, as defined by total function $u : M \rightarrow U$, where U is a universe of measure units.*
3. *a distributive aggregation function, as defined by total function $f : M \rightarrow \{\textsc{Sum}, \textsc{Max}, \textsc{Min}\}$.*
4. *an asserted value (primary fact), as defined by partial function $v : M \rightarrow V$. A measure $m \in M$ has an asserted value iff the connection-level of m is equivalent to the top-connection-level of r, i.e.: $v(m)$is defined $\Leftrightarrow b(m) = \hat{l}_r$.*

When talking about different m-relationships, e.g. r and r', we alternatively use subscripts (e.g., M_r and $M_{r'}$) or quotes (e.g. M, b, being features of r and M', b' being features of r') to denote the context of sets and functions.

Definition 5 (Measure Units and Measure Types)**.** *Each measure unit $u \in U$ is member of one measure type $t \in T$, where T is a universe of measure types, as defined by total function type : $U \rightarrow T$.*

Example 6 (M-Relationship)**.** Consider m-relationship *sales* in Fig. 4 between m-objects *Product*, *Time*, and *Location*. It defines measure *revenue* at connection-level ⟨*model,month,city*⟩ with unit of measure € and aggregation function Sum.

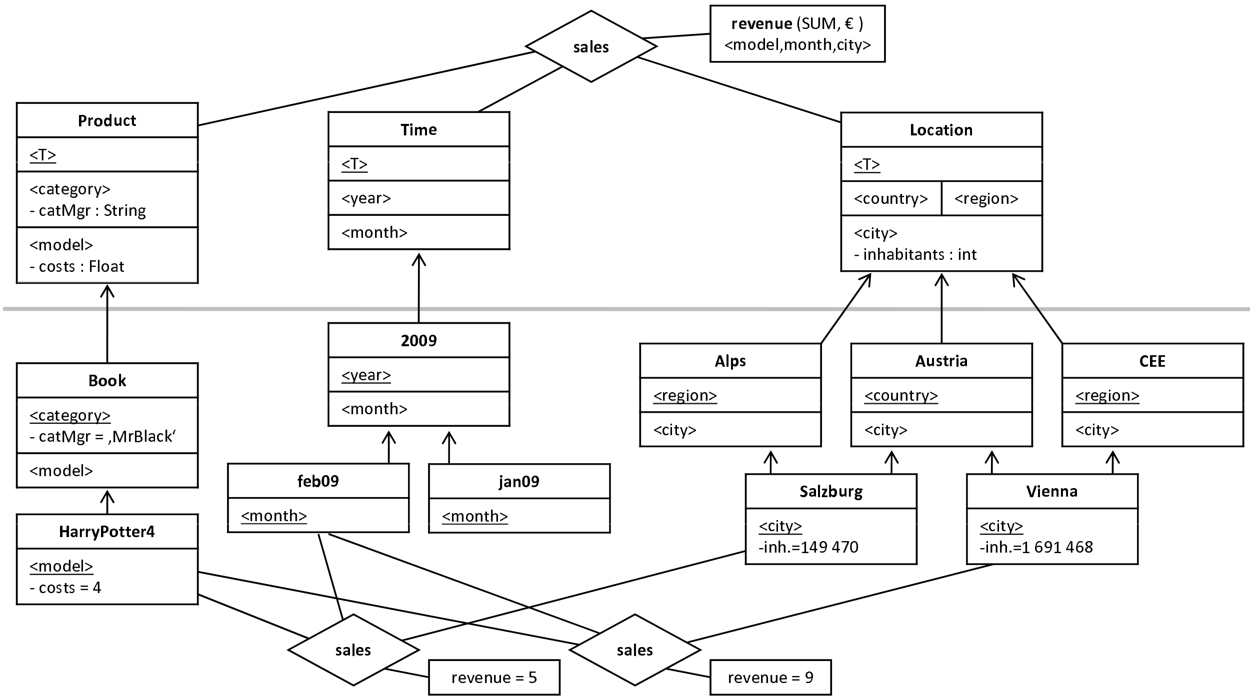

Figure 4: Homogeneous data warehouse modeled with m-objects and m-relationships

To discuss concretization of m-relationships we need the notions of *partial order of connection levels* and *partial order of coordinates.*

Definition 6 (Partial Order of Connection Levels)**.** *Given a coordinate* $(o_1, ..., o_n)$ *and the levels of the m-objects of that coordinate,* $L_{o_1}, ..., L_{o_n}$*, and two connection-levels* $(l'_1, ..., l'_n), (l_1, ..., l_n) \in (L_{o_1} \times ... \times L_{o_n})$*. We say* $(l'_1, ..., l'_n)$ *is a* descendant *of* $(l_1, ..., l_n)$*, written as* $(l'_1, ..., l'_n) \preceq (l_1, ..., l_n)$*, iff for i=1..n each level* l'_i *is a descendant of* l_i*, i.e.,*
$(l'_1, ..., l'_n) \preceq (l_1, ..., l_n) \Leftrightarrow l'_1 \preceq l_1 \wedge ... \wedge l'_n \preceq l_n$.

Definition 7 (Partial Order of Coordinates)**.** *Given n dimensions,* $D1, ..., Dn$*, of n disjoint sets of m-objects,* $O_{D1}, ..., O_{Dn}$*). We say coordinate* $(o'_1, ..., o'_n) \in (O_{D1} \times ... \times O_{Dn})$ *is a* descendant of or equal to *coordinate* $(o_1, ..., o_n) \in (O_{D1} \times ... \times O_{Dn})$*, written as* $(o'_1, ..., o'_n) \preceq (o_1, ..., o_n)$*, iff* $\forall_{i=1}^n : (o'_i, o_i) \in H_Di^*$*. In this case we also speak of a* sub-coordinate*.*

Coordinate $(o'_1, ..., o'_n)$ *is a* descendant of *- or* proper sub-coordinate *of - coordinate* $(o_1, ..., o_n)$*, written as* $(o'_1, ..., o'_n) \prec (o_1, ..., o_n)$*, iff for all dimensions i=1..n,* o'_i *is a descendant of or is equal to* o_i*, and for at least one dimension j,* o'_j *is a concretization of* o_j*, i.e.,* $\forall_{i=1}^n : (o'_i, o_i) \in H^*_{Di} \wedge \exists_{j=1}^n : (o'_j, o_j) \in H^+_{Dj}$*.*

Coordinate $(o'_1, ..., o'_n)$ overlaps with *coordinate* $(o_1, ..., o_n)$*, written as* $(o'_1, ..., o'_n) \between (o_1, ..., o_n)$*, iff they have some (sub-)coordinates in common, that is, for all dimensions,* $i = 1..n$*, the respective dimension m-objects* o'_i, o'_i *are either equal or in a concretization relationship:* $(o'_1, ..., o'_n) \between (o_1, ..., o_n) \Leftrightarrow (\forall_{i=1}^n : (o'_i, o_i) \in H^*_{Di} \vee (o_i, o'_i) \in H^*_{Di})$

An m-relationship is concretized by substituting one or more of the m-objects in its coordinate by descendant m-objects. The descendant m-relationship must provide values for the measures at its top-connection-level and may add measures, and move the connection-level of a measure to a more specific connection-level.

Definition 8 (Consistent Concretization of M-Relationships)**.** *A m-relationship* $r' = (o'_1, ..., o'_n; M', b', u', f', v') \in R$ *is a consistent concretization of another m-relationship* $r = (o_1, ..., o_n; M, b, u, f, v) \in R$*, iff*

1. $(o'_1, ..., o'_n) \prec (o_1, ..., o_n)$
2. *every measure m of r,* $m \in M$*, with a base-level that is below or equal to the top-level of* r'*, is also a measure of* r'*, every other measure of r is not a measure of* r' (measure containment)*:*
   $\{m \in M \mid b(m) \preceq \hat{l}_{r'}\} \subseteq M'$
   $\{m \in M \mid b(m) \not\preceq \hat{l}_{r'}\} \cap M' = \emptyset$
3. *for each measure m shared by r and* r'*, the base-level of m at* r' *is the same or below the base-level of m at r:* $\forall m \in (M \cap M') : b'(m) \preceq b(m)$ (assured granularity)
4. *Common measures are associated with measure units of the same measure type and the same aggregation function: For* $m \in (M \cap M')$*:*
   (a) *type*$(u'(m))$ = *type*$(u(m))$ (stability of measure types)
   (b) $f'(m) = f(m)$ (stability of aggregation functions)

Example 7 (Concretization of M-Relationships)**.** M-relationship *sales* between m-objects *HarryPotter4*, *feb09*, and *Salzburg* concretizes *sales* between m-objects *Product*, *Time*, *Location* and its top-connection level is ⟨*model,month,city*⟩, thus it defines a value for measure *revenue*. An example for introducing additional levels and moving measures to more specific connection-levels will be given later.

3.2 Modeling Hetero-Homogeneous M-Cube-Hierarchies with M-Relationships

We first describe how a homogeneous cube of n dimensions can be modeled by m-relationships: (1) A cube is represented by a concretization hierarchy of

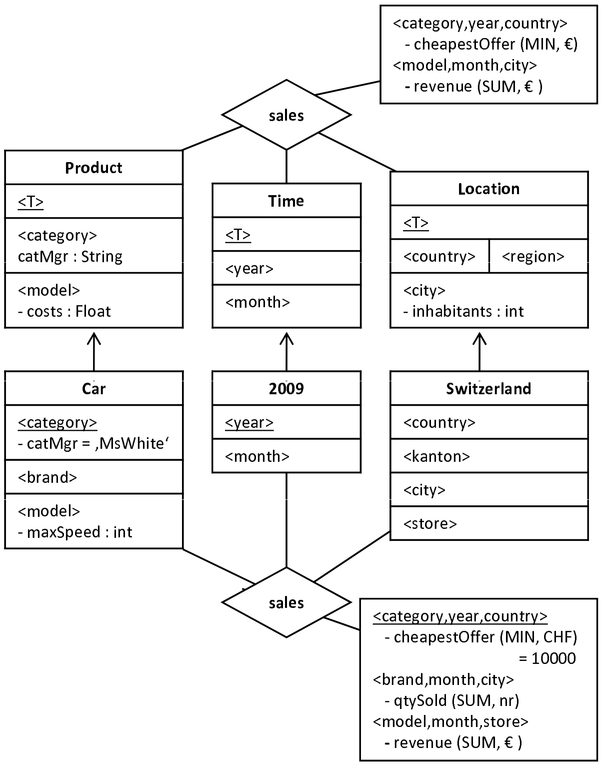

Figure 5: Concretization of m-relationship sales, also representing a hetero-homogeneous cube

n-ary m-relationships. (2) The root m-relationship connects the root m-objects of these n dimensions. (3) The root m-relationship has measures associated with a single connection-level which consists of the bottom levels of these n dimensions and gives the measures of the cube. (4) The cells or facts of the cube are represented by m-relationships that concretize the root m-relationship and connect n m-objects that are at the connection-level for which the measures of the root-m-relationship are defined and give values for these measures.

Example 8 (Homogeneous Cube)**.** Fig. 4 depicts a homogeneous cube schema *sales* (above the gray horizontal line) and its facts (below the gray line). Note, while the m-cube approach provides a coherent model both for cube- and dimension-schemas as well as their instances, its graphical representation is obviously not meant to be used to fully model a cube with all its facts and dimension instances; it is rather used, analogously to object diagrams in UML, to model exemplary dimension instances and facts together with dimension and cube schema. The cube schema corresponds to the Dimensional Fact Model depicted in 1. The cube extension has two facts.

We now describe how a hetero-homogeneous cube of n dimensions can be modeled by m-relationships: Cubes can be hetero-homogeneous in that (1) sub-cubes have *additional measures*, (2) different sub-cubes may give *different units* for the same measure (3) various measures are provided at *mixed granularities*(4) different sub-cubes may provide the same measure at *different granularities* (see examples given in the Introduction).

Additional Measures can be introduced at a sub-cube identified by a coordinate $(o_1, ..., o_n)$ as follows: An m-relationship for this coordinate is introduced. This m-relationship defines a measure for the connection-level at which values for the measure are provided.

Different sub-cubes with *different units* for the same measure are supported as follows: An m-relationship for the coordinates of each sub-cube is introduced and gives a different unit of measure.

Cubes with measures that are provided at *mixed granularities* can be represented as follows: An m-relationship is introduced that associates these measures with different connection levels.

Cubes in which different sub-cubes provide the same measure at *different granularities* are represented as follows: An m-relationship is introduced for the cube and gives measure at some connection-level. For each sub-cube that provides this measure at a more detailed granularity, an m-relationship is introduced and associates this measure with a more specific connection level.

Example 9 (Hetero-Homogeneous Cubes)**.** Fig. 5 depicts a fragment of a hetero-homogeneous cube. - Note, this example is different from previous ones for sake of presentation and simplicity. - M-relationship *sales* between m-objects *product*, *time*, and *location* introduces two measures at mixed granularities. Measure *cheapestOffer* for connection-level $\langle$*category,year,country*$\rangle$ and measure *revenue* for connection-level $\langle$*model,month,city*$\rangle$. M-relationship *sales* between category *car*, year *2009*, and country *Switzerland* concretizes the above m-relationship as follows: (1) It introduces an additional measure *qtySold* for connection-level $\langle$*model,month,city*$\rangle$. (2) It moves measure *revenue* from connection-level $\langle$*model,month,city*$\rangle$ to $\langle$*model,month,store*$\rangle$. Thus it provides for different granularity of measure *revenue*: the cube will have stored revenue values for models of cars, months in 2009, and stores in Switzerland, but not for other product categories, months in other years, stores in other countries. (3) It provides a different unit of measure for *cheapestOffer*, that is swiss francs instead of €.

The notion of a multi-level cube (m-cube), as defined below, generalizes the cube in the Dimensional Fact Model (Golfarelli et al., 1998).

Definition 9 (Multi-Level Cube)**.** *A multi-level cube* $C = (D_1, ..., D_n; S, R)$ *connects* n *dimensions,* $D_1, ..., D_n$*. Its root-coordinate* S *is identified by a tuple* $(o_1, ..., o_n) \in O_{D_1} \times ... \times O_{D_n}$*.* R *is a set of m-relationships which represent the measure schema and the base facts of* C*.*

The m-relationships of C that provide a measure-value are called the base facts or base cells of C.

When talking about different m-cubes, e.g. C and C', we alternatively use subscripts (e.g., X_C) or quotes (e.g. D_1, S, being features of C and D'_1, S' being features of C') to denote the context of sets and functions. Whenever the context is clear we use unquoted variables (e.g. D_1, S).

We now define consistency criteria that avoid conflicts due to multiple inheritance and avoid overlapping facts. For this definition we use the set $\hat{R}_r$ of *directly subsuming m-relationships* of a m-relationship r, $\hat{R}_r := \{r' \in R \mid r \preceq r' \wedge \nexists r'' \in R : r \preceq r'' \prec' r\}$.

Definition 10 (Consistent M-Cube)**.** *A multi-level cube* $C = (D_1, ..., D_n; S, R)$ *with root-coordinate* $S = (o_1, ..., o_n)$ *is consistent iff*

1. *there is one m-relationship in* R *that corresponds to root-coordinate* S*.*
2. *for each cell* $x \in X$ *there is at most one corresponding m-relationship in* R*.*
3. *For each pair of m-relationships* $r, r' \in R$*, if* r' *is a concretization of* r*,* $r' \preceq r$*, then* r' *is a consistent concretization of* r *according to Def. 8.*

4. *Each measure is introduced at only one m-relationship:* $\forall r, r' \in R : \exists m \in \{M_r \cap M_{r'}\} \Rightarrow \exists r'' \in R : m \in M_{r''} \wedge coord(r) \preceq coord(r'') \wedge coord(r') \preceq coord(r'')$ (unique induction rule for measures)

5. *For each measure m shared by two overlapping m-relationships r and r', $coord(r) \between coord(r')$, if r defines a value for m than r' must not define a value for m: $v_r(m)$is defined $\Rightarrow v_{r'}$ is not defined.* (unique assertion of values)

6. *For each non-empty cell x, for each pair r, r' of direct subsuming m-relationships of x that contain a measure m with base-level below or equal to the level of x, the measure unit and the base level for m are the same at r and r':*
$\forall x \in X, \forall r, r' \in \hat{R}_x, \forall m \in M_r \cap M_{r'}$:
$(\exists r \in R : coord(r) \preceq x) \wedge (b_r(m) \preceq \hat{l}_x \vee b_{r'}(m) \preceq \hat{l}_x)) \Rightarrow$
   (a) $u_r(m) = u_{r'}(m)$ (unit conflict avoidance)
   (b) $b_r(m) = b_{r'}(m)$ (base level conflict avoidance)

Unit conflict avoidance and base level conflict avoidance (Def. 10, item 6) ensure that possible conflicts due to multi-dimensional concretization are solved explicitly by an m-relationship directly beneath the conflicting m-relationships.

An m-cube represents hetero-homogeneous base facts as possibly extracted and loaded from various source OLTP databases. An m-cube defined between with root-coordinate $(o_1, ..., o_n)$ implicitly also represents a cube of cubes. This cube of cubes consists of a set of homogeneous cubes, one for each n-tuple of levels in the cartesian product of the levels of the m-objects of the root coordinate. The cells of such a cube are given by the cartesian product of m-objects at those levels.

Example 10. Fig. 6 depicts the homogeneous cubes of m-cube *sales*; Fig. 7 shows a sample whereby we ignore dimension *time* for simplicity.

A hetero-homogeneous cube exists for each sub-coordinate and consists of those m-relationships of the given cube that are descendants of that sub-coordinate.

Example 11. Sub-m-cube *sales(Car, Time, Switzerland)* takes a closer look at car sales in Switzerland. Fig. 8 depicts the homogeneous cubes of this sub-m-cube ignoring dimension time. Note, that the dimension levels identifying these cubes are not shown. Additional cubes become available for the additional dimension levels *kanton* and *store* defined for country *Switzerland* (see Fig. 3) and additional level *brand* defined for category *car*. Further, additional measure *qtySold* is available for the cubes at connection-level ⟨*brand,city*⟩ and above, since this measure has been defined for cars in Switzerland for this level (see Fig. 5). Note that for descendant connection-levels the cubes show a null-value for this measure.

The aggregate cell (or fact) has the coordinate of the m-cube and a value for each measure that is provided for this coordinate or can be calculated from the base cells of the m-cube.

Example 12. The top-left entry in Fig. 7 and the top-left entry in Fig. 8 represent the aggregate cells of coordinates *(product,location)* and *(car,switzerland)* respectively.

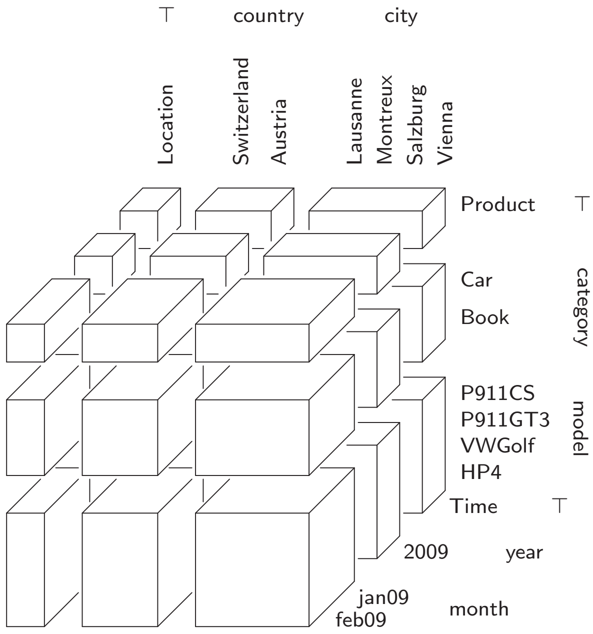

Figure 6: Visualization of homogeneous cubes derived from dimensions *product*, *time*, and *location* of m-cube *sales*. For simplicity, dimension level *region* is not shown

Thus, a multi-level cube implicitly describes all (roll-up) cubes that can be derived from a base cube and its dimensions.

In the subsequent section we introduce an m-cube algebra whose operations can be used to extract the aggregate cell, a homogeneous cube, or a hetero-homogeneous cube from an m-cube. There we also describe how the measure values of these cubes and the aggregate cell (fact) are determined.

4 Querying M-Cubes

In this section we introduce an algebra for multi-level cubes. There are three types of operators

1. closed m-cube operators (*dice* δ, *slice* σ, *import-union* $\cup_i$, *projection* π) apply to m-cubes and produce m-cubes as result

2. the *fact extraction* operator φ applies to an m-cube and extracts all measure values of a given cell into a relation with a single tuple

3. the (roll-up) *cube* operator κ applies to an m-cube and produces a homogeneous roll-up cube (a relation with primary and/or aggregated facts) as result, which allows to apply traditional cube operations and facilitates integration with current data warehouse technology

Some common cube-operations like *roll-up* and *drill-down* are not part of the algebra but are defined as mappings from one κ-application to another κ-application.

Based on this m-cube algebra we propose a two-stage approach to analyze data in hetero-homogeneous m-cubes, (1) *selecting a (sub-)m-cube* and (2) specifying the *query* based on the schema of the selected (sub-)m-cube. Queries consistent with the schema of the (sub-)m-cube return homogeneous and correct answers. Note that a sub-m-cube typically has a richer schema than a more general m-cube (as exemplified in Fig. 8).

		⊤	country		city			
		Location	Switzerland	Austria	Lausanne	Montreux	Vienna	Salzburg
⊤	Product	20	13	7	6	7	7	4
category	Car	13	10	3	5	5	3	4
	Book	7	3	4	1	2	4	4
	P911CS	7	5	2	3	2	2	4
model	P911GT3	7	5	2	3	2	2	4
	VWGolfXY	6	5	1	2	3	1	4
	HP4	7	3	4	1	2	4	4

Figure 7: Homogeneous cubes of a sample m-cube *sales* showing values for measure *revenue* as defined by m-relationship *sales* between *Product* and *Location*, ignoring dimension *Time* and level *region*.

	country	kanton	city		store		
	Switzerland	Vaud	Lausanne	Montreux	tellInc	gesslerLtd	TschudiComp
Car	28/17	28/17	18/11	10/6	9/	9/	10/
P911	19/10	19/10	11/**6**	8/**4**	5/	6/	8/
VWGolf	9/7	9/7	7/**5**	2/**2**	4/	3/	2/
P911CS	10/	10/	5/	5/	**3**/	**2**/	**5**/
P911GT3	9/	9/	6/	3/	**2**/	**4**/	**3**/
VWGolfXY	9/	9/	6/	3/	**4**/	**3**/	**2**/

Figure 8: Homogeneous cubes of sub-m-cube with root-coordinate (*Car*,*Switzerland*) of m-cube *sales* depicting measures *revenue* and *qtySold* where available

The *query* consists of (i) optionally a set of boolean *predicates* to narrow the analysis on cells whose m-objects fulfil the predicate (corresponds to operation *slice* in Def. 15) (ii) optionally a set of *measures of interest* (corresponds to operation *projection* in Def. 12), (iii) optionally a *measure unit* for each measure and (iv) a *cell coordinate* to retrieve facts of a single cell, or a *cube coordinate* to retrieve facts of all cells within the specified cube (corresponds to operations *fact extraction* in Def. 20, and *cube extraction* in Def. 22, respectively). If not specified explicitly all available measures are considered and values are converted to the measure unit specified at the specified (sub-)m-cube (see Fig. 9 for an example query and its results).

4.1 Closed M-Cube Operations

The *dice*-operator selects a sub-m-cube from an m-cube.

Definition 11 (Dice δ)**.** *Given an input m-cube $C = (D_1, ..., D_n, S, R)$, coordinate $(o_1, ..., o_n)$, and that there is a m-relationship $r = (o_1, ..., o_n, M, b, u, f, v) \in R$, then $\delta_{o_1,...,o_n} C$ results in output-cube $C' = (D_1, ..., D_n, S', R')$ with $S' = (o_1, ..., o_n)$*
$R' = \{r' \in R \mid coord(r') \preceq (o_1, ..., o_n)\}$

Example 13. Dice operation $\delta_{(Car,2009,Switzerland)} sales$ retrieves a sub-m-cube *car09SalesCH* containing m-relationships with coordinates that are descendants of *(Car,2009,Switzerland)*. Fig. 8 depicts the cube of cubes of this m-cube.

	tellInc	gesslerLtd
P911CS	3	2
P911GT3	2	4
VWGolfXY	4	3

Figure 9: Homogeneous cube of sales revenue for car sales in 2009 in Switzerland in big cities with cells at level $\langle model, Time, store \rangle$

The projection operator applied on an m-cube, returns an m-cube with a reduced set of measures.

Definition 12 (Projection π)**.** *Given an input m-cube $C = (D_1, ..., D_n, S, R)$, and a set of measures $\mathcal{M} \in M_C$, then $\pi_{\mathcal{M}} C$ results in output-cube $C' = (D_1, ..., D_n, S, R')$, with R' defined as follows: for each $r = (o_1, ..., o_n; M, b, u, f, v) \in R$ there is a $r' = (o_1, ..., o_n; M', b', u', f', v') \in R'$, with $M' := M \cap \mathcal{M}$, and for each $m \in M'$: $b'(m) := b(m)$, $u'(m) := u(m)$, $f'(m) := f(m)$, $f'(m) := f(m)$, and $v'(m) := v(m)$.*

As prerequisites for predicates used as selection criteria in slice-operation we define the notions of stable upward navigation and class extension.

Definition 13 (Upward Navigation)**.** *The ancestor m-object of m-object $o \in O_D$ at level $l \in L_o$, denoted as $o[l]$, is defined by*
$o[l] \stackrel{\text{def}}{=} o' : (o, o') \in H_D^* \wedge \hat{l}_{o'} = l.$

An m-object represents for each level of direct or indirect descendants the class of descendant m-objects of that level. To refer to the set of m-objects at level l beneath m-object o, we write $o\langle l \rangle$. For example, $car\langle model \rangle$ refers to the set of m-objects at level *model* beneath m-object *Car*.

Definition 14 (Class Extension)**.** *The class of m-objects of m-object $o \in O_D$ at level $l \in L_o$, denoted as $o\langle l \rangle$, is defined by*
$o\langle l \rangle \stackrel{\text{def}}{=} \{o' \mid (o', o) \in H_D^* \wedge \hat{l}_{o'} = l\}.$

A *predicate* is a boolean expression over attributes of a class of m-objects, $o\langle l \rangle$ and of its ancestors (using upward navigation). Note, that predicates could be predefined at m-objects and associated with a level like attributes. Then these predicates could be overwritten in concretizations.

A *slice*-operation on a given m-cube selects all coordinates at a given level that fulfill the given criteria and returns an m-cube with all m-relationships from the given m-cube that are between descendants of the given coordinates, between ancestors of the given coordinates, or are at these coordinates. Dimensions $D_1, ..., D_n$ and root-coordinate $S = (o_1, ..., o_n)$ are the same in both the input m-cube and the output m-cube. An outer-slice has the same output but additionally consists of all m-relationships from the input m-cube that are above the cube-level of the selection.

Definition 15 (Slice σ)**.** *Given are an input m-cube $C = (D_1, ..., D_n, S, R)$ with $S = (o_1, ..., o_n)$ and selection predicates $(p_1, l_1), ..., (p_n, l_n)$. For $\sigma_{(p_1,l_1),...,(p_n,l_n)}$ to be applicable on C, there must not be an m-relationship in R with an asserted measure value above cube-level $(l_1, ..., l_n)$. The slice operation $\sigma_{(p_1,l_1),...,(p_n,l_n)} C$ results in output cube $C' = (D_1, ..., D_n, S, R')$ where R' is given as follows. Let the selected cells be given by $\bar{X} := \{o \in o_1\langle l_1 \rangle \mid p_1(o)\} \times ... \times \{o \in o_n\langle l_n \rangle \mid p_n(o)\}$; and let the included m-relationships be given by $\bar{R} := \{r \in R \mid \exists x \in \bar{X} : r \preceq x\}$. Then $R' := \bar{R} \cup \{r \in R \mid \exists \bar{r} \in \bar{R} : \bar{r} \preceq r\}$.*

Example 14 (Slice). Slice-operation $\sigma_{(inhabitants>100000,city)}$ *car09SalesCH* selects m-cube *car09SalesCHinBigCities*, which comprises m-relationships representing car sales of 2009 in Switzerland in cities with more than 100000 inhabitants.

Definition 16 (Outer Slice $\bar{\sigma}$). *Outer Slice is defined as Slice in Def. 15 with the difference that R' is defined as follows:*
$R' := \bar{R} \cup \{r \in R \mid (l_1, ..., l_n) \preceq \hat{l}_r\}$

Import Union inserts a cube into an existing cube. It can be seen as a bulk operation for inserting m-relationships. The resulting cube needs to be consistent according to Def. 9.

Definition 17 (Import Union $\cup_i$). *Given two input cubes, main cube $C = (D_1, ..., D_n, S, R)$ and to-be-imported cube $C' = (D_1, ..., D_n, S', R')$, with $\nexists r \in R : r \preceq S'$ and $S' \preceq S$, then $C \cup_i C'$ results in output cube $C'' = (D_1, ..., D_n, S, R'')$ with $R'' := R \cup R'$.*

4.2 Fact and Cube Extraction

Before defining fact and cube extraction operators we need to investigate which measures are available for a given coordinate. A measure at a given coordinate may be provided by a m-relationship of the m-cube, i.e., be an asserted fact, or be derived through application of the aggregation function provided with the measure definitions.

Definition 18 (Common Measures at Coordinates). *Given a coordinate $x = (o_1, ..., o_n)$ from a consistent m-cube $C = (D_1, ..., D_n; S, R)$, its set of measures, M_x, is given by the union of measures of its direct subsuming m-relationships $\hat{R}_x$, given that the measures connection-level is below or equal to the level of x: $M_x :=$*
$\{m \in \bigcup_{r \in \hat{R}_x} M_r \mid \forall r \in \hat{R}_x : b_r(m) \preceq (\hat{l}_{o_1}, ..., \hat{l}_{o_n})\}$

For each measure $m \in M_x$, given one of its direct subsuming m-relationships $r \in \hat{R}_x$ that contains m, $m \in M_r$, the base-level, unit-of measure, and aggregation function are those defined at r:

1. $b_x(m) := b_{r'}(m)$
2. $u_x(m) := u_{r'}(m)$
3. $f_x(m) := f_{r'}(m)$

Conversion between measure units is facilitated by multi-polymorphic function *conv*. It applies, dependent on the pair of source and target measure units, a simple arithmetic expression on the numeric input value to produce an output value. We assume, that there is a conversion expression for each pair of measure units that are members of the same measure type. Context-sensitive unit conversion, e.g. time-dependent currency conversion, is facilitated by extending function *conv* to take dimension objects, i.e. a cell-coordinate, as additional parameters. The extended *conv*-method is multi-polymorphic in the two measure-units and in these dimension-objects. For space-limitations we do not further discuss this extension and refer the interested reader to (Schrefl et al., 1998).

Given a source measure unit $u_s \in U$, a target measure unit $u_t \in U$, with $type(u_s) = type(u_t)$, and an input value $v \in V$, operation $conv(u_s, u_t, v)$ returns a value that is the conversion of value v from measure unit u_s to measure unit u_t.

We now define how measure values are derived from asserted facts.

Definition 19 (Aggregation of Measures *val*). *Given an m-cube $C = (D_1, ..., D_n, S, R)$, a cell $x = (o_1, ..., o_n)$ with $\exists r \in R : r \preceq x$, measure $m \in M_x$, and measure unit $u \in U$, with $type(u) = type(u(m))$, then the value of measure m at coordinate x converted to unit u, $val(m, x, u)$, is calculated by applying aggregation function $f_x(m)$ on the set of converted m-values of m-relationships below or at cell x, given by $R_x := \{r \in R \mid r \preceq x\}$; or null if this set is empty, i.e.:*

$$val(m, x, u) := \begin{cases} f_x(m)(\bigcup_{r \in R_x} conv(& \text{if } \exists r \in R_x : \\ \quad u_r(m), u, (v_r(m)))) & \quad (v_r(m) \text{ is defined}) \\ null & \text{otherwise} \end{cases}$$

We are now ready to define the fact extraction operator.

Definition 20 (Fact Extraction φ). *Given an m-cube $C = (D_1, ..., D_n, S, R)$, a cell $x = (o_1, ..., o_n)$ with $\exists r \in R : r \preceq x$, and a mapping from measures to measure units $(m_1 \mapsto u_1, ..., m_k \mapsto u_k)$, then fact extraction operation $\varphi_{(o_1,...,o_n),(m_1 \mapsto u_1,...,m_k \mapsto u_k)}C$ returns a relation with schema $(D_1, ..., D_n, m_1 : u_1, ..., m_k : u_k)$ and an instance consisting of one tuple $(o_1, ..., o_n, val(m_1, x, u_1), ..., val(m_k, x, u_k))$.*

When leaving out the mapping from measures to measure units, fact extraction results in a relation with all measures that are available at the respective cell and converts each measure to the respective unit of measure defined at this cell (see Def. 21).

Definition 21 (Fact Extraction Shorthand φ). *Given cell $x = (o_1, ..., o_n)$ with $\exists r \in R : r \preceq x$ with measures $M_x = \{m_1, ..., m_k\}$ and units of measures $u_x = \{m_1 \mapsto u_1, ..., m_k \mapsto u_k\}$, then $\varphi_{o_1,...,o_n}C$ is a shorthand for $\varphi_{(o_1,...,o_n),(m_1 \mapsto u_1,...,m_k \mapsto u_k)}C$.*

A cube extraction operation returns a homogeneous cube, consisting of a tuple for each non-empty cell at a given cube-level.

Definition 22 (Cube Extraction κ). *Given an m-cube $C = (D_1, ..., D_n, S, R)$, a cube-level $(l_1, ..., l_n) \in (L_{D_1}, ..., L_{D_n})$, and a mapping from measures to measure units $(m_1 \mapsto u_1, ..., m_k \mapsto u_k)$.*

The set of non-empty cells of C at level $(l_1, ..., l_n)$, denoted as $C\langle l_1, ..., l_n\rangle$, is given by $\{(o_1, ..., o_n) \in X \mid (\exists r \in R : r \preceq (o_1, ..., o_n)) \wedge \hat{l}_{o_1} = l_1 \wedge ... \wedge \hat{l}_{o_1} = l_1\}$

The result of cube extraction operation $\kappa_{(l_1,...,l_n),(m_1 \mapsto u_1,...,m_k \mapsto u_k)}C$ is the relation given by union of facts of all non-empty cells at level $(l_1, ..., l_n)$:
$\kappa_{(l_1,...,l_n),(m_1 \mapsto u_1,...,m_k \mapsto u_k)}C :=$
$\bigcup_{x \in C\langle l_1,...,l_n\rangle}(\varphi_{x,(m_1 \mapsto u_1,...,m_k \mapsto u_k)}C)$

Example 15. Given our m-cube *car09SalesCHinBigCities* of car sales, the homogeneous cube with measure *revenue* of sales rolled up to level *model,store* can be extracted by applying projection and subsequent cube extraction operators, e.g., $\kappa_{(model,store),(revenue \mapsto €)}\pi_{revenue}$ *car09SalesCHinBigCities*. Fig. 9 depicts the result of this query as cross table.

In order to retain measures that are available at some but not all cells of a cube, we use outer union (Codd, 1979) on facts extracted according to Def. 21. Note that we accept null values and heterogenous measure units in the resulting cube (see Def. 23).

Definition 23 (Outer Cube Extraction $\bar{\kappa}$). *The result of $\bar{\kappa}_{(l_1,...,l_n)}C$ is the relation given by outer union, denoted as $\bar{\bigcup}$, on facts of all non-empty cells, at level $(l_1, ..., l_n)$:*
$\bar{\kappa}_{(l_1,...,l_n)}C := \bar{\bigcup}_{x \in C\langle l_1,...,l_n\rangle}(\varphi_c C)$

5 Related Work

Heterogeneities in data warehouses are widely acknowledged as an important research direction and have received considerable attention in the literature, especially on data warehouse integration (Torlone, 2008; Berger and Schrefl, 2008), summarizability (Hurtado and Mendelzon, 2001), OLAP visualization (Mansmann and Scholl, 2006; Cuzzocrea and Mansmann, 2009), and conceptual modeling (Malinowski and Zimányi, 2006). These works especially discuss heterogeneities in dimension hierarchies, such as non-covering, non-strict, and asymmetric hierarchies. However, to the best of our knowledge, none of these approaches provides for a top-down modeling approach of hetero-homogeneous dimension and cube hierarchies.

Conceptual data warehouse design has attracted a lot of work, various approaches are based on entity-relationship modeling, such as (Song et al., 2008), on the UML, such as (Trujillo et al., 2001), or on abstract state machines (Zhao and Schewe, 2004). The well-established Dimensional Fact Model (Golfarelli et al., 1998) has been used in this paper as starting point to illustrate homogeneous data warehouse schemas and how hetero-homogeneous hierarchies extend them.

An important area of work concerns summarizability (Lenz and Shoshani, 1997; Hurtado and Mendelzon, 2001) and formal aspects of aggregation in data warehouses (Lenz and Thalheim, 2001). In this context (Gray et al., 1997) introduce the notions of distributive, algebraic, and holistic aggregation functions. In this paper we only considered measures based on distributive aggregation functions, a restriction we will relax in future work.

6 Conclusion

In this paper we introduced hetero-homogeneous hierarchies and discussed their application to data warehousing. We provided structural definitions and consistency criteria based on m-objects and m-relationships.

We believe that hetero-homogeneous hierarchies are a very promising approach to modeling and querying data warehouses. Interesting issues which we will investigate in the future are:

- *Aggregation operations.* In this paper we limited the discussion on measures based on distributive aggregation functions SUM, MAX, MIN. We excluded operation COUNT due to the lack of a meaningful definition of its semantics in the presence of different and mixed granularities. Future work needs to address peculiarities of aggregation operations in multi-level cubes, in the flavor of (Lenz and Thalheim, 2001), especially concerning empty cells, as well as algebraic and holistic aggregation operations.

- *Prototype.* Future work needs to provide a proof-of-concept prototype. We will investigate how our m-cube approach can be implemented on top of object-relational DBMS.

- *Efficiency.* In this paper we discussed a conceptual modeling and querying approach, disregarding optimization issues. In the future we also want to investigate how hetero-homogeneous hierarchies can be implemented and queried efficiently.

References

Abelló, A., Samos, J. and Saltor, F. (2006), YAM^2: a multidimensional conceptual model extending UML, *Inf. Syst.* **31**(6), 541–567.

Berger, S. and Schrefl, M. (2008), From federated databases to a federated data warehouse system, *HICSS 2008*.

Codd, E. F. (1979), Extending the database relational model to capture more meaning, *ACM Trans. Database Syst.* **4**(4), 397–434.

Cuzzocrea, A. and Mansmann, S. (2009), OLAP visualization: Models, issues, and techniques, *in* J. Wang, ed., 'Encyclopedia of Data Warehousing and Mining, Second Edition', Information Science Reference.

Golfarelli, M., Maio, D. and Rizzi, S. (1998), The dimensional fact model: A conceptual model for data warehouses, *Int. J. Cooperative Inf. Syst.* **7**(2-3), 215–247.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F. and Pirahesh, H. (1997), Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub totals, *Data Min. Knowl. Discov.* **1**(1), 29–53.

Hurtado, C. A. and Mendelzon, A. O. (2001), Reasoning about summarizability in heterogeneous multidimensional schemas, *ICDT 2001*, pp. 375–389.

Lenz, H.-J. and Shoshani, A. (1997), Summarizability in OLAP and statistical data bases, *SSDBM 1997*, pp. 132–143.

Lenz, H.-J. and Thalheim, B. (2001), OLAP databases and aggregation functions, *SSDBM 2001*, IEEE Computer Society, pp. 91–100.

Malinowski, E. and Zimányi, E. (2006), Hierarchies in a multidimensional model: From conceptual modeling to logical representation, *Data Knowl. Eng.* **59**(2), 348–377.

Mansmann, S. and Scholl, M. H. (2006), Extending visual OLAP for handling irregular dimensional hierarchies, *in* A. M. Tjoa and J. Trujillo, eds, 'DaWaK', Vol. 4081 of *Lecture Notes in Computer Science*, Springer, pp. 95–105.

Neumayr, B., Grün, K. and Schrefl, M. (2009), Multi-level domain modeling with m-objects and m-relationships, *APCCM 2009*.

Schrefl, M., Kappel, G. and Lang, P. (1998), Modeling collaborative behavior using cooperation contracts, *Data Knowl. Eng.* **26**(2), 191–224.

Song, I.-Y., Khare, R., An, Y., Lee, S., Kim, S.-P., Kim, J. and Moon, Y.-S. (2008), Samstar: An automatic tool for generating star schemas from an entity-relationship diagram, *ER 2008*, pp. 522–523.

Torlone, R. (2008), Two approaches to the integration of heterogeneous data warehouses, *Distributed and Parallel Databases* **23**(1), 69–97.

Trujillo, J., Palomar, M., Gómez, J. and Song, I.-Y. (2001), Designing data warehouses with OO conceptual models, *IEEE Computer* **34**(12), 66–75.

Zhao, J. and Schewe, K.-D. (2004), Using abstract state machines for distributed data warehouse design, *APCCM 2004*, pp. 49–58.