Hierarchy integration in the design of data warehouses

Yasser Hachaichi
Mir@cl Laboratory
ISAAS, B.P.1013
3018 Sfax, Tunisia
+yasser.hachaichi@fsegs.mn.tn

Jamel Feki
Mir@cl Laboratory
FSEG, B.P.1088
3018 Sfax, Tunisia
+jamel.feki@fsegs.mn.tn

ABSTRACT

Face to their merger and/or collaboration with partners, today’s enterprises often need to integrate several databases. As a result, their decision making process ends up analyzing data coming from various databases. While database integration has been thoroughly examined, it is only recently that the integration of multidimensional models has drawn attention. A multidimensional model is a data model that facilitates the analysis operations during the decision making process. It organizes data into facts that can be analyzed according to dimensions represented through hierarchies. This paper presents an integration process for the hierarchy concept. In particular, it proposes a set of basic integration operations and constraints that produce a multidimensional model that is loadable from different data sources. It illustrates the integration process through examples.

Keywords
Data Warehouse, Hierarchies Integration, Multidimensional Schemas.

1. INTRODUCTION

Information systems of organizations are generally divided into two classes: Operational systems and decision support systems. Operational systems serve the need of running the daily activities of the business, whereas decision support systems provide historical information to trace and analyze the business in order to make judicious business decisions [1]. In fact, many companies have realized the importance of the information treasure hidden within their operational systems, which can significantly improve their decision quality [2].

Data warehousing is a technology that: collects all relevant data into one central system; organizes the data efficiently so it is consistent and easy to retrieve; and keeps “old” data for historical analyses [3]. William Inmon, who coined the term “data warehouse”, defined a data warehouse as “a subject oriented, integrated, nonvolatile, and time variant collection of data in support of management decision” [4]. Nowadays, many data warehouses are developed using the dimensional modeling approach; this model can be composed of a set of coherent data marts (DM) [5] each of which provides a dimensional view of a single business process.

Since the DW and/or the DM are loaded with the data generated by the operation support system, several researchers proposed dimensional modeling approaches for the most commonly used data source models [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. In our previous works, we have defined three automatic bottom-up/data-driven methods for DM schemas design. These methods start respectively from relational database [9], XML documents data-centric [14] and object oriented databases [15]. Our methods exploit recent versions of a data source in the operation support systems and automatically apply a set of rules that extract all multidimensional concepts (facts with their measures, dimensions and hierarchies) and then produce DM schemas dedicated to business decision.

Overall, similar to our approach, current propositions treat each data model separately. However, due to economic constraints, an enterprise is often forced to open its operational system to collaborate with partners and/or to ensure the collaboration among its affiliates whose data models might be different. Hence, one needs a way to integrate the DMs produced by the approach appropriate to the data source model. Instead of integrating data sources with different structures, we believe that the integration of DMs is easier since DMs are structured in a uniform and simple way, along the widely accepted notions of dimension and fact [16].

The main objective of this paper is to propose a process for DM integration in order to design DM schemas loadable from different sources. As a second objective, the proposed process can also be used to build multidimensional patterns [17] that are generic DM schemas used to assist a decision maker to specify his/her analytical needs. In addition, the integration of multidimensional data (i.e., DMs) is a reasonable solution for the problem of building a complete DM gathering data from several existing DMs.

The problem of integration has been studied in the literature extensively for heterogeneous databases [18], [19], [20], [21], [22]. This problem presents, in general many facets and, a survey of the solutions proposed are studied in [23], [22]. In this paper, we take apart the general aspects of the problem, and we do not address the general issues of the integration problem, such as the automatic matching of terms [24]; rather, we focus our attention on the structural integration of hierarchies. Hierarchies represent a main component in a DM schema as they are crucial for DrillDown and RollUp OLAP (On-Line Analytical Processing) operations. Indeed, the hierarchies structure distinguish multidimensional schemas from databases and their integration raises specific issues not yet addressed by the data warehousing community.

This paper is organized as follows. Section 2 overviews the main concepts of multidimensional models. Section 3 presents the metrics, operations and constraints we use to define an integration process in Section 4. Finally, the paper is summarized and its future work is outlined in Section 5.
2. MULTIDIMENSIONAL MODEL

As argued in, the traditional conceptual models for databases, such as the Entity-Relationship model or UML, do not provide a suitable means to describe the fundamental aspects of data warehouses. One crucial need in DW is to represent explicitly certain important characteristics of the information contained therein, which are not related to the abstract representation of real world concepts, but rather to the final goal of the data warehouse: supporting data analyses oriented to decision making. More specifically, it is widely recognized that there are at least two specific notions (i.e., concepts) that any conceptual data model for DW should include: fact and dimension. A dimension represents a business perspective under which data analysis is to be performed and is organized in a hierarchy of levels, which correspond to different ways to group its fact elements (i.e., measures). A fact represents factual data on which the analysis is focused and associates measures with coordinates, defined over a set of dimension levels [25].

In order to clearly and accurately define our proposed method for hierarchy integration, we first formally introduce our notation of the multidimensional concepts [26] namely fact, measure, dimension and hierarchy.

Definition 1. A fact $F$ is defined as $(\text{name}, \text{Mf})$ where:
- $\text{name}$ is the name of the fact,
- $\text{Mf}=\{m_1, m_2, \ldots, m_B\}$ is a finite set of measures,
  where each measure $m_i$ is defined as $m_i=(\text{NameMf}_i, \text{FuncMf}_i)$
- $\text{NameMf}_i$ is the name of a measure
- $\text{FuncMf}_i$ is an aggregate function (Sum, Average, ..).

Definition 2. A dimension $d$ is defined as $(d^d, \text{Att}, \text{HIER})$ where:
- $d^d$ is the name of the dimension,
- $\text{Att}$ is a set of all attributes of $d$ (including weak attributes),
- $\text{HIER}=(H_1^d, H_2^d, \ldots, H_p^d)$ is a set of all hierarchies of $d$.

Definition 3. A hierarchy $H_i^d$ of a dimension $d$ is an acyclic path defined as $(N_i^d, \text{ParamF}, \text{AttF})$ where:
- $N_i^d$ is the name of the hierarchy,
- $\text{ParamF}=\{p_1, p_2, \ldots, p_n, \text{All}\}$ is an ordered list of $n+1$ attributes used in $H_i^d$. These attributes are usually called parameters;
- $\{p_i, p_j\}$ denotes a partial order called roll up relation on the parameters; it means that $p_i$ rolls up to $p_j$.
- the $(n+1)^{th}$ parameter, called $\text{All}$, enables to calculate most aggregated values.
- $\text{AttF}$ is a function that associates an attribute $p_i$ to a set of its weak-attributes with $\forall i \in \{1, \ldots, n\}$, $\text{AttF}(p_i)=\{a_{t_1}, \ldots, a_{t_n}\}$ and $\forall j \in [c..r], a_{t_j} \in \text{Att} \text{ et } a_{t_j} \notin \text{ParamF}$.

We are now ready to introduce the keystone concept of multidimensional modeling namely the multidimensional schema which is composed of one or more facts containing measures, and a collection of dimensions made up of attributes organized into hierarchies. Within a multidimensional schema, each fact is linked to a subset of dimensions.

Note that a multidimensional schema can be either a star schema analysing a single fact examined according to its dimensions or a constellation schema gathering several facts with shared dimensions [10]. Each schema belongs to one specific application domain. In the reminder of this paper, we use multidimensional schema as a generic term for both star and constellation schemas.

Definition4. A multidimensional schema is defined as a tuple $(Nsch, Fsch, DIM, \text{Func})$ where:
- $Nsch$ is the name of the multidimensional schema,
- $N^{sch}_{dim}$ is the name of a domain to which the schema belongs,
- $Fsch=(F1, F2, \ldots, Fs)$ is a set of facts,
- $\text{DIM}=[d_1, d_2, \ldots, d_r]$ is a set of dimensions,
- $\text{Func}$ is a function which associates a fact $F_i$ to the list of its dimensions with $\forall i \in [1..s]$, $\text{Func}(F_i)$ = $\{d_1, \ldots, d_r\}$ with $\forall j \in \{i..p\}, d_j \in \text{DIM}$.

Multidimensional schemas can also be represented more naturally as diagrams according to the DFM model [27]. Figure 1 shows an example of two multidimensional schemas $S1$ and $S2$. Each schema models the Sales fact with measures Qty_Sold and Price. These measures are recorded according to the dimensions: DATE, CLIENT, PRODUCT, RETAIL_OUTLET and STORE.

Figure 1. Two multidimensional schemas with heterogeneous hierarchies.

3. METRIC, OPERATIONS AND CONSTRAINTS FOR HIERARCHY INTEGRATION

Similar to most database schema integration approaches, we first define a comparison metric between hierarchies; it allows us to calculate the similarity between two autonomous hierarchies to be integrated. Secondly, we define the necessary operations to perform this integration and, finally we propose a set of constraints to produce consistent merged hierarchies.

3.1 Hierarchy comparison

Given two hierarchies belonging to two different multidimensional schemas, to decide whether they should be integrated, we examine their common parameters. We consider only parameters because weak attributes are less significant and unused during OLAP analysis; in fact, they are generally served for labeling results.

In order to compare two hierarchies $H_i^{d1}$ and $H_i^{d2}$ belonging to two different multidimensional schemas, we define the metric $\Psi$ which calculates the similarity between parameters $(\text{ParamF})$ of $H_i^{d1}$ and those of $H_i^{d2}$:

$$\Psi (H_i^{d1}, H_i^{d2}) = \frac{|H_i^{d1}.\text{ParamF} \cap H_i^{d2}.\text{ParamF}|}{\text{Dim}(H_i^{d1}.\text{ParamF}) \cap \text{Dim}(H_i^{d2}.\text{ParamF})}$$

where:
- $|H_i^{d1}.\text{ParamF}|$ is the number of parameters in $H_i^{d1}$.
Note that common parameters can be identified using linguistic relationships of standard techniques for semantic reconciliation [22]. Therefore, we suppose the existence of the Boolean function $Eqv(pi, pj)$ that returns True if the names of the two parameters $pi$ and $pj$ are identical or synonyms, and False otherwise.

We use the $\Psi$ metric to evaluate the semantic similarity ratio between two hierarchies. The more this ratio is important, the more the integration of these hierarchies is significant. Hence, we distinguish five different cases among which only three recommend the hierarchy integration. To illustrate these cases, we define the five following functions:

1. **Idt ($H_{d1}^j, H_{d2}^j$):** means that the two hierarchies have only parameters with equivalent names. This occurs when: $\Psi(H_{d1}^j, H_{d2}^j) = 1 \land |H_{d1}^j.ParamF\cap H_{d2}^j.ParamF| > 0.4$

2. **Incl ($H_{d1}^j, H_{d2}^j$):** indicates that all parameters of one hierarchy are included within those of the other hierarchy. This happens when: $\Psi(H_{d1}^j, H_{d2}^j) = 1 \land |H_{d1}^j.ParamF\cap H_{d2}^j.ParamF| > 0.4$

3. **SInt ($H_{d1}^j, H_{d2}^j$):** denotes that there exists a “relatively important” intersection between the parameters of the two hierarchies. This arises when: $0.4 \leq \Psi(H_{d1}^j, H_{d2}^j) < 1$

Note that the value 0.4 is a threshold that could be set by the designer according their experience.

4. **Whit ($H_{d1}^j, H_{d2}^j$):** it occurs when: $0 < \Psi(H_{d1}^j, H_{d2}^j) < 0.4$

5. **Disj ($H_{d1}^j, H_{d2}^j$):** means that there is no relationship between the two hierarchy parameters: $\Psi(H_{d1}^j, H_{d2}^j) = 0$

When one of the first three above functions is satisfied, we proceed to the integration through the operations we define in the next section.

### 3.2 Integration operations

For the integration process we define the set of the six operations below.

#### 3.2.1 Hierarchy deletion (DelH)

This operation deletes a hierarchy $h$ from a dimension $d$:

**Syntax:**

\[
\text{DelH (Sch, d, h)} = \text{Sch}
\]

**Input:**
- $\text{Sch} = (N_{sch}, N_{sch}, F_{sch}, DIM, \text{Funct})$
- $d = (N_\text{d}, \text{Att}, \text{HIER})$ is a dimension,
- $h = (N_h, \text{ParamF}, \text{AttF})$ the hierarchy to delete with $h.\text{ParamF} = \langle p_1, p_2, \ldots, p_n \rangle$

**Conditions:**
- $d \in \text{DIM}$, $h \subset d.\text{HIER}$

**Schema changes:**
- $d.\text{Att} = d.\text{Att} \setminus h.\text{ParamF} \cup h.\text{ParamF}$
- $d.\text{HIER} = d.\text{HIER} \setminus h$

**Output:** $\text{Sch}$ is the input multidimensional schema without the hierarchy $h$ in the dimension $d$.

#### 3.2.2 Sub Hierarchy deletion (DelSub_H)

This operation deletes a sub-hierarchy from a hierarchy $h$ on a dimension $d$:

**Syntax:**

\[
\text{DelSub_H (Sch, d, h, l1, ln)} = (\text{Sch, h'})
\]

**Input:**
- $\text{Sch} = (N_{sch}, N_{sch}, F_{sch}, DIM, \text{Funct})$
- $d = (N_\text{d}, \text{Att}, \text{HIER})$ is a dimension
- $h = (N_h, \text{ParamF}, \text{AttF})$ is a hierarchy with $h.\text{ParamF} = \langle p_1, p_2, \ldots, p_n \rangle$
- $l_1, l_n$ delimits the sub-hierarchy to be deleted from $h$

**Conditions:**
- $d \in \text{DIM}$, $h \subset d.\text{HIER}$
- $1 \leq l_1, l_n \leq |h.\text{ParamF}|$

**Schema changes:**
- $h' = (N_h', \text{ParamF}, \text{AttF})$
- $h'.\text{ParamF} = \langle p_{l_1}, \ldots, p_{l_n} \rangle$
- $h.\text{ParamF} = \langle p_1, p_2, \ldots, p_n \rangle$
- $d.\text{Att} = d.\text{Att} \cup \langle p \in h.\text{ParamF} \setminus (\text{AttF}(p) \cup \text{AttF}(\langle p \rangle)) \rangle$

**Output:** $h'$ is the deleted sub-hierarchy made up of attributes of $h$ from $l_1$ to $l_n$

Sch is the input multidimensional schema from which the sub-hierarchy $h'$ is deleted from the dimension $d$ and the hierarchy $h$ is adjusted.

Note that we preserve the deleted sub-hierarchy $h'$ in the output because it may be inserted elsewhere using ConSub_H.

#### 3.2.3 Sub Hierarchy Connection (ConSub_H)

This operation connects a sub-hierarchy $h'$ at the position $l$ to a hierarchy $h$ of a dimension $d$, thus creating a new hierarchy $h''$ on $d$.

**Syntax:**

\[
\text{ConSub_H (Sch, d, h, l, h')} = \text{Sch}
\]

**Input:**
- $\text{Sch} = (N_{sch}, N_{sch}, F_{sch}, DIM, \text{Funct})$
- $d = (N_\text{d}, \text{Att}, \text{HIER})$ is a dimension
- $h$ a hierarchy with $h.\text{ParamF} = \langle p_1, \ldots, p_n \rangle$
- $h'$ a hierarchy to insert in $d/h.\text{ParamF} = \langle p_1, \ldots, p_n \rangle$
- $l$ the level at which $h'$ will be added to $h$

**Conditions:**
- $d \in \text{DIM}$, $h \subset d.\text{HIER}$
- $h' \subset d.\text{HIER}$

**Schema changes:**
- $d.\text{Att} = d.\text{Att} \cup h.\text{ParamF} \cup \langle p \in h.\text{ParamF} \setminus \text{AttF}(p) \rangle$
- $d.\text{HIER} = d.\text{HIER} \cup h''$
- $h''.\text{ParamF} = \langle p_1, \ldots, p_L, p_{L+1}, \ldots, p_n \rangle$

**Output:** $\text{Sch}$ is a multidimensional schema where the dimension $d$ is enriched with the new hierarchy $h''$.

#### 3.2.4 Linking parameters (AddLink)

This operation links two parameters located at positions $L1$ and $L2$ in a hierarchy $h$ of a dimension $d$.

**Syntax:**

\[
\text{AddLink (Sch, d, h, L1, L2)} = \text{Sch}
\]

**Input:**
- $\text{Sch} = (N_{sch}, N_{sch}, F_{sch}, DIM, \text{Funct})$
- $d = (N_\text{d}, \text{Att}, \text{HIER})$ is a dimension

**Output:** $\text{Sch}$ is the input multidimensional schema without the hierarchy $h$ in the dimension $d$.  

---

*Annual International Academic Conference on Business Intelligence and Data Warehousing (BIDW 2010)*
- $h = (N^h, \text{Param}F, \text{Att}F)$ is a hierarchy with $h.\text{Param}F = \langle p_1, ..., p_L \rangle$.

**Conditions:**
- $d \in \text{DIM}$, $h \subset d.\text{HIER}$
- $1 \leq L1 \leq L2 \leq |h.\text{Param}F|$

**Schema changes:**
- $d.\text{HIER} = d.\text{HIER} \cup \{h'\}$ where $h'.\text{Param}F = \langle p_1, ..., p_{L1}, p_{L1+1}, ..., p_{L2} \rangle$

**Output:**
$\text{Sch}$ is the input multidimensional schema in which the parameter $p$ is inserted at position $L$ of the hierarchy $h$.

These integration operations should produce meaningful hierarchies, i.e., hierarchies that can be loaded from the source and exploited during OLAP analyses (data should be rolled up correctly through hierarchies). In the next section, we address this problem by setting constraints on hierarchies coming from multiple sources.

### 3.3 Hierarchy integration constraints
Since our objective is to integrate hierarchies coming from multiple data sources, we need to set up specific constraints. These latter deal with non common parameters, trace the complete path of a hierarchy by data source and introduce artificial identifiers for hierarchy dimensions when necessary.

#### 3.3.1 Hierarchy disjunction constraint
In multidimensional data models, data are viewed as points in a multidimensional space. Each dimension could be represented as a directed graph of its parameters with a parent/child relationship among them. An important application of this structure is to use it to infer data summarizability; that is, whether an aggregate view defined for some parameters can be correctly derived from a set of pre-computed views defined for other parameters[28].

Integrating heterogeneous hierarchies issued from distinct dimensions produces a dimension noted $d$, $d$ may be structurally heterogeneous if two parameters (i.e., issued from different hierarchies) at a given level are allowed to have two immediate parents at two different levels. **Hierarchy disjunction constraint**, allows us to reason about summarizability in such heterogeneous dimensions. This constraint was addressed by [28] to respect summarizability in dimensions with multiple hierarchies to analyze data exclusively according to one hierarchy.

We adopt the **hierarchy disjunction constraint** as restriction statements that indicate legal analysis paths for measures coming from different data sources. As an example, consider the Product dimension of Figure 2. If we want to impose that measures aggregated along the hierarchy containing Branch parameter (loaded from source $S1$) must not be aggregated along the hierarchy containing Department parameter (and vice versa), then we must define a disjunction constraint; this latter states that each product has one direct parent in either the path ($\langle \text{Id}_P.\text{Branch} \rangle, S1$) or in the path ($\langle \text{Id}_P.\text{Department} \rangle, S2$) exclusively. The symbol $\otimes$ denotes this constraint as shown in Figure 2.

Note that we aim at analyzing data from multiple sources so that each path of hierarchy of parameters may be loaded from a specific source. Thus, we associate each hierarchy with its loading data source ($S$).

In the conceptual model, we graphically represent this constraint by linking such paths with $\otimes$ and annotating each side of the constraint symbol with its appropriate data source.
thus, we propose one value for each different source. Thanks to dimension rows by different values of the dimension from multiple sources, we propose to distinguish the artificial parameter. However, since we may load an integrated single data source, then we need a unique value for the parameter. When the content (i.e., country) within a hierarchy produces as many rows as there are different values for this level (i.e., values by country). To aggregate further these rows to a single one, an artificial level called 'All' (with the unique value 'all' in all rows of the dimension) is generally appended on top of the most general level (e.g., country); this aggregation collapses all rows into a single row. Thus the 'All' parameter represents the ancestor parent node within any dimension; in other terms, it is the node that connects all top parameters of hierarchies of any dimension.

When the content (i.e., rows) of a dimension is loaded from a single data source, then we need a unique value for the 'All' artificial parameter. However, since we may load an integrated dimension from multiple sources, we propose to distinguish the dimension rows by different values of the 'All' artificial attribute; thus, we propose one value for each different source. Thanks to this distinction, we can trace the origin of the data and, therefore prepare to correctly rollup data by source. For example in figure 8 AllS1 concerns values loaded from S1 and AllS2 concerns the values loaded from S2.

3.3.4 Artificial identifier for an integrated dimension
In multidimensional models, all hierarchies in a dimension d have the identifier of d as their first common finest parameter (i.e., this constraint is known as hierarchical root [30]). Obviously, this constraint is generally unsatisfied when similar hierarchies issued from two different dimensions have to be integrated. In order to guarantee this hierarchical root constraint during the integration of such hierarchies, we propose to add an artificial identifier (noted Id_Art) as the identifier for the integrated dimension. In practice, this identifier is a sequential integer. For example an artificial identifier is added to the dimension d2 of Figure 11 because this dimension is the result of the integration of d1 and d2 of figure 10 which have different identifiers (respectively B and A).

4. HIERARCHY INTEGRATION PROCESS
Recall that the hierarchy integration process merges two heterogeneous hierarchies and traces the data origin. In addition, the similarity metric (cf. section 3.1) identifies five semantic relationships; only three of them are significant for the integration: Idt, Incl and Snt. The two others (Wnt and Disj) correspond to few or zero common parameters between the hierarchies, therefore we voluntarily neglect them in our integration process.

The integration process for the case identity of hierarchies is trivial. For each of the Incl and the Snt, we define three cases according to the location of the common parameters: at the beginning, in the middle (i.e., anywhere between the first and last parameter), or at the end. These three basic locations cover the majority of situations. However, other complex and less frequent situations that combine the basic cases could be encountered. Due to space limitation, they are not discussed in this paper.

In the remainder of this section, we focus on the previous seven elementary cases of merging two hierarchies H(Hsch1, d1, Hi) belonging to different schemas Sch1 and Sch2. We first introduce each case through an illustrative example and then, we specify the required sequence of operations to perform the integration.

4.1 Identity of hierarchies
The identity of two hierarchies H(Hsch1, d1, Hi) and H(Hsch2, d2, Hj) means that they are of the same length (i.e., same number of parameters) and their entire parameters are equivalent in respect to their positions (cf. Figure 4). The integration process applies the next two operations:

- UnionWeak_H(Sch2, d2, Hj, Sch1, d1, Hi)
- DelH (Sch1, d1, Hj)
As shown in Figure 5, this integration produces the hierarchy $H^d_2$ enriched with the weak attributes of $H^d_1$ and removes $H^d_1$ from $d1$ in $Sch1$.

4.2 Inclusion of hierarchies

The inclusion of hierarchy $H^d_1$ in $H^d_2$ (noted $H^d_1 \subset H^d_2$) means that the set of all parameters of $H^d_1$ is included in the parameters of $H^d_2$. In this case, the integration steps depend on the location of the parameters common between $H^d_1$ and $H^d_2$. This location may be at the beginning, in the middle or at the end. In this section we detail how we process each of these three cases and we use the two functions $pl(H)$ and $ph(H)$ to get the position of the lowest and highest common parameters in the hierarchy $H$, respectively.

4.2.1 All common parameters at the beginning

As Figure 7 illustrates, the sequence of all parameters of $H^d_1$ $<$A, B, C> is included at the beginning of $H^d_2$. Thus, the integration consists mainly of merging weak attributes and specifying the all parameters by hierarchy data source. This integration process can be done through the following operations:

- $UnionWeak_H$(Sch2, d2, $H^d_2$, Sch1, d1, $H^d_1$)
- $DelH$(Sch1, d1, $H^d_1$)
- $AddLevAt$(Sch2, d2, $H^d_2$, |$H^d_1$.PramF|+1, AllS2)
- $AddPara$(Sch2, d2, AllS2, $H^d_2$, ph($H^d_2$)+1, AllS1)
- $AddPara$(Sch2, d2, $H^d_2$, 1, Id_Art)
- $AddLink$(Sch, d, h, 1, $pl(H^d_2)$+1)

The above operations should respect the following constraint:

- $(C, AllS1, S1) \ominus (C, D, S2)$

As a result, this integration enriches the hierarchy $H^d_2$ by the weak attributes issued from $H^d_1$ (cf. figure 8)
In Dimensions Date of figure 1 there is an inclusion in the middle. The integration result of these dimensions is given in figure 12.

As depicted in Figure 13, all common parameters of $H_{d1}^{Hi}$ are at the end of $H_{d2}^{Hj}$. To analyze data according to these hierarchies with respect to their sources, we introduce an artificial identifier that becomes the root node for the two merged hierarchies. Moreover, in order to preserve measure summarizability, we add the disjunction hierarchy constraint. Hence, the integration process proceeds as follows:

- $\text{UnionWeak}_H(\text{Sch}_2, d_2, H_{d2}^{Hj}, \text{Sch}_1, d_1, H_{d1}^{Hi})$
- $\text{AddPara}(\text{Sch}_2, d_2, \text{Sch}_1, d_1, H_{d1}^{Hi})$
- $\text{AddPara}(\text{Sch}_2, d_2, \text{Sch}_1, d_1, H_{d2}^{Hj})$
- $\text{AddLink}(\text{Sch}_2, d_2, H_{d2}^{Hj}, 1, \text{Id}_\text{Art})$

under the following constraint

- $(\text{Id}_\text{Art}, A, S2) \otimes (\text{Id}_\text{Art}, C, S1)$

The result of this integration is the hierarchy $H_{d2}^{Hj}$ augmented with the weak attributes of common parameters in $H_{d1}^{Hi}$ (cf. Figure 14).

4.2.3 All common parameters at the end

Figure 12. Date integration result

Figure 13. Inclusion and common parameters at the end.

4.3 Strong Intersection

**Strong Intersection** means that there are several parameters common between the two hierarchies but without any inclusion, i.e., each hierarchy has at least one specific parameter.

Once again, the integration process of two hierarchies $H_{d1}^{Hi}$ and $H_{d2}^{Hj}$ depends on the three possible locations (begin, middle or end) of their common parameters. In this section, we study how to integrate hierarchies in each case and we conventionally produce the result in $H_{d2}^{Hj}$.

4.3.1 All common parameters at the beginning

Figure 15 shows an example for the strong intersection between two hierarchies where all common parameters locate at the beginning of $H_{d1}^{Hi}$ and $H_{d2}^{Hj}$. In this case, the integration produces two hierarchies those share the common parameters and split at their highest common one (i.e., parameter B); each hierarchy ends with its original uncommon parameters as shown in Figure 15.

To perform this integration, we apply the following sequence of operations:

- $\text{UnionWeak}_H(\text{Sch}_2, d_2, H_{d2}^{Hj}, \text{Sch}_1, d_1, H_{d1}^{Hi})$
- $\text{AddPara}(\text{Sch}_2, d_2, \text{Sch}_1, d_1, H_{d1}^{Hi})$  
- $\text{AddPara}(\text{Sch}_2, d_2, \text{Sch}_1, d_1, H_{d2}^{Hj})$
- $\text{AddLink}(\text{Sch}_2, d_2, H_{d2}^{Hj}, 1, \text{Id}_\text{Art})$

under the following constraint

- $(\text{Id}_\text{Art}, A, S2) \otimes (\text{Id}_\text{Art}, C, S1)$

The constraint to be satisfied is:

- $(B, D, S1) \otimes (B, C, S2)$

Figure 14. Integration result: Inclusion and common parameters at the end.

Figure 15. Strong intersection and common parameters at the beginning.

Figure 16. Integration result: Strong intersection and common parameters at the beginning.
4.3.2 All common parameters in the middle
When the parameters shared between $H_{d1}^{d2}$ and $H_{d2}^{d1}$ are located inside, the integration adds an artificial identifier, creates two hierarchies that meet in the lowest common parameter, and splits at the highest common parameter. In order to trace the uniform constraint two exclusion constraints are added.

Figure 17. Strong intersection and common parameters in the middle.

The example of Figure 17 will be integrated according to the following operations:
- $\text{UnionWeak}_H(Sch2, d2, H_{d2}^{d1}, Sch1, d1, H_{d1}^{d2})$
- $\text{AddPara}(Sch2, d2, AllS2, H_{d2}^{d1}, |H_{d2}^{d1}.PramF|+1)$
- $\text{AddPara}(Sch1, d1, AllS1, H_{d1}^{d2}, |H_{d1}^{d2}.PramF|+1)$
- $\text{DelSub}_H(Sch1, d1, H_{d1}^{d2}, p_l(H_{d1}^{d2}), |H_{d1}^{d2}.PramF|+1) = (Sch', h')$
- $\text{ConSub}_H(Sch2, d2, H_{d2}^{d1}, p_l(H_{d2}^{d1}), h')$
- $\text{DelH}(Sch1, d1, H_{d1}^{d2})$
- $\text{AddPara}(Sch2, d2, H_{d2}^{d1}, 1, \text{Id}_A) \text{ under the respect of following constraint}$
  - $(\text{Id}_A, W, S1) \otimes (\text{Id}_A, A, S2)$

in respect to the three following constraints and produces the result depicted in figure 18.

- $(\text{Id}_A, W, S1) \otimes (\text{Id}_A, A, S2)$
- $(D, Y, S1) \otimes (D, E, S2)$
- $(\text{Id}_A, W, S1) \otimes (\text{Id}_A, A, S2) \otimes (D, Y, S1) \otimes (D, E, S2)$

Figure 18. Integration result: Strong intersection and common parameters in the middle.

The application of this case on dimensions STORE of figure 1 is depicted on figure 19.

Figure 19. STORE integration result.

4.3.3 All common parameters at the end
When the two hierarchies to be integrated have all their common parameters at the end (Figure 20), the integration process builds two hierarchies with an artificial identifier linked to the original identifier, adds the All attribute for each hierarchy and associates correspondent constraints (Figure 21).

Figure 20. Strong Intersection and common parameters at the end.

The example of Figure 20 will be integrated according to the following operations:
- $\text{UnionWeak}_H(Sch2, d2, H_{d2}^{d1}, Sch1, d1, H_{d1}^{d2})$
- $\text{AddPara}(Sch2, d2, AllS2, H_{d2}^{d1}, |H_{d2}^{d1}.PramF|+1)$
- $\text{AddPara}(Sch1, d1, AllS1, H_{d1}^{d2}, |H_{d1}^{d2}.PramF|+1)$
- $\text{DelSub}_H(Sch1, d1, H_{d1}^{d2}, p_l(H_{d1}^{d2}), |H_{d1}^{d2}.PramF|+1) = (Sch', h')$
- $\text{ConSub}_H(Sch2, d2, H_{d2}^{d1}, p_l(H_{d2}^{d1}), h')$
- $\text{DelH}(Sch1, d1, H_{d1}^{d2})$
- $\text{AddPara}(Sch2, d2, H_{d2}^{d1}, 1, \text{Id}_A) \text{ under the respect of following constraint}$
  - $(\text{Id}_A, W, S1) \otimes (\text{Id}_A, A, S2)$

Dimensions PRODCUT of figure 1 correspond to the case: Strong Intersection with common parameters at the end. The result of their integrations is presented in figure 21.

Figure 21. PRODUCT integration result.

5. CONCLUSION

In this paper, we proposed a set of operations and constraints that provides for the integration of hierarchies in order to produce a multidimensional model loadable from different sources. For a better understanding, and eventually the automation, of the proposed operations, we formalized the proposed operations. In addition, we illustrated their application through an example.

We are in the process of integration these operations with our CASE toolset [31] for the design of multidimensional schemas from heterogeneous data sources. This automation will allow us to evaluate the applicability of the operations. Furthermore, we
are currently defining the integration process for the cases where the common parameters are scattered within the hierarchies being integrated; this process will rely on integration processes for the basic cases (at the beginning, in the middle and at the end) which we have presented in this paper.

6. REFERENCES


