

Extending Visual OLAP for Handling Irregular Dimensional Hierarchies

Svetlana Mansmann and Marc H. Scholl

University of Konstanz, P.O. Box D188, 78457 Konstanz, Germany
{Svetlana.Mansmann, Marc.Scholl}@uni-konstanz.de

Abstract. Comprehensive data analysis has become indispensable in a variety of environments. Standard OLAP (On-Line Analytical Processing) systems, designed for satisfying the reporting needs of the business, tend to perform poorly or even fail when applied in non-business domains such as medicine, science, or government. The underlying multidimensional data model is restricted to aggregating only over summarizable data, i.e. where each dimensional hierarchy is a balanced tree. This limitation, obviously too rigid for a number of applications, has to be overcome in order to provide adequate OLAP support for novel domains.

We present a framework for querying complex multidimensional data, with the major effort at the conceptual level as to transform irregular hierarchies to make them navigable in a uniform manner. We provide a classification of various behaviors in dimensional hierarchies, followed by our two-phase modeling method that proceeds by eliminating irregularities in the data with subsequent transformation of a complex hierarchical schema into a set of well-behaved sub-dimensions.

Mapping of the data to a visual OLAP browser relies solely on meta-data which captures the properties of facts and dimensions as well as the relationships across dimensional levels. Visual navigation is schema-based, i.e., users interact with dimensional levels and the data instances are displayed on-demand. The power of our approach is exemplified using a real-world study from the domain of academic administration.

1 Introduction

Data warehouse technology, initially introduced in the early 90s to support data analysis in business environments, has recently become popular in a variety of novel applications like medicine, education, research, government etc. End-users interact with the data using advanced visual interfaces that enable intuitive navigation to the desired data subset and granularity as well as its expressive presentation using a wide spectrum of visualization techniques.

Data warehouse systems adopt a *multidimensional data model* tackling the challenges of the On-Line Analytical Processing (OLAP) [2] via efficient execution of queries that aggregate over large data volumes. Analytical values within this model are referred to as *measures*, uniquely determined by descriptive values drawn from a set of *dimensions*. The values within a dimension are typically organized in a containment type hierarchy to support multiple granularities.

Standard OLAP ensures correct aggregation by enforcing *summarizability* in all dimensional hierarchies. The concept of *summarizability*, first introduced in [10] and further explored in [5] and [3], requires distributive aggregate functions and dimension hierarchy values, or informally, it requires that 1) facts map directly to the lowest-level dimension values and to only one value per dimension, and 2) dimensional hierarchies are balanced trees [5].

At the level of visual interfaces, summarizability is also crucial for generating a proper navigational hierarchy. Data browsers present each hierarchical dimension as recursively nested folders allowing users to browse either directly in the dimensional data, in which case each hierarchical entity can be expanded to see its child values, or in the dimensional attributes, where each hierarchical level is mapped to a sub-folder of its parent level's folder. Simple OLAP tools, e.g., Cognos PowerPlay [1], tend to provide only the data-based navigation whereas advanced interfaces, such as Tableau Software [13] and SAP NetWeaver BI [11], combine schema navigation with data display. Figure 1 shows the difference between data- and schema-based browsing for a hierarchical dimension Period.

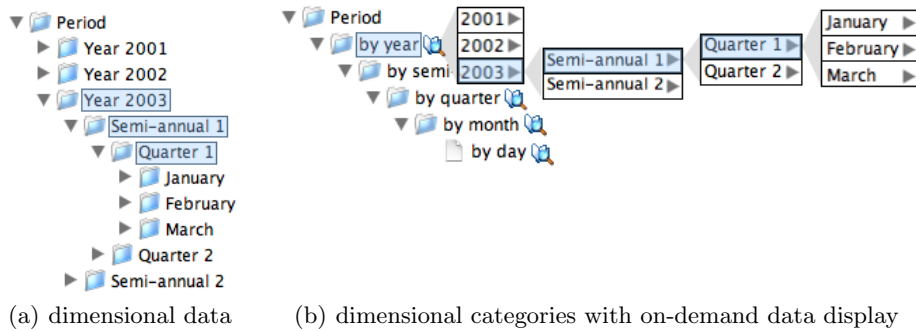


Fig. 1. Browsing in dimensional hierarchies: data vs. schema navigation

Analysts are frequently confronted with non-summarizable data which cannot be adequately supported by standard models and systems. To meet the challenges of novel applications, OLAP tools are to be extended at virtually all levels of the system architecture, from conceptual, logical and physical data transformation to adequately interfacing the data for visual querying and providing appropriate visualization techniques for comprehensive analysis.

This paper presents an OLAP framework capable of handling a wide spectrum of irregular dimensional hierarchies in a uniform and intuitive manner. All introduced extensions are supported by enriching the meta-data and providing algorithms for interfacing the data and mapping user interaction back to OLAP queries. The remainder of the paper is structured as follows: Section 2 sets the stage by describing related work and a motivating real-world case study from the area of academic administration. A classification of supported hierarchical patterns and re-modelling techniques for heterogeneous hierarchies are presented in Section 3, followed by the methods for data transformation and translating

the multidimensional schema into a navigational framework in Section 4. We summarize our contribution and identify future research directions in Section 5.

2 Motivation and Related Work

2.1 Related Work on Multidimensional Data Modelling

A number of researchers have recognized the deficiencies of the traditional OLAP data model [15] and suggested a series of extensions at the conceptual level.

A powerful approach to modeling dimension hierarchies along with SQL query language extensions called $SQL(\mathcal{H})$ was presented in [4]. $SQL(\mathcal{H})$ does not require data hierarchies to be balanced or homogeneous. Niemi et al. [6] analyzed unbalanced and ragged data trees and demonstrated how dependency information can assist in designing summarizable hierarchies. Hurtado et al. [3] propose a framework for testing summarizability in heterogeneous dimensions.

Pedersen et al. have formulated further requirements an extended multidimensional data model should satisfy and evaluated 14 state-of-the-art models from both the research community and commercial systems in [9]. Since none of the existing models was even close to meeting most of the defined requirements, the authors proposed an extended model for capturing and querying complex multidimensional data. This model, supporting non-summarizable hierarchies, many-to-many relationships between facts and dimensions, handling temporal changes and imprecision, is by far the most powerful multidimensional data model of the current state of the art. A prototypical implementation of an OLAP engine called the Tree Scape System, which handles irregular hierarchies by normalizing them into summarizable ones, is described in [8].

To our best knowledge most of the extensions formalized by the above models have not been incorporated into any visual OLAP interface. In our previous work [14] we presented some insights into visual querying of heterogeneous and mixed-granularity dimensions. Our current contribution is an attempt to further reduce the gap between powerful concepts and deficient practices by designing a comprehensive framework for visual analytical querying of complex data.

2.2 Motivating Case Study

Our presented case study is concerned with the accumulated data on the expenditures within a university. Academic management wishes the data to be organized into an OLAP cube where the fact table *Expenditures* contains single orders with the measure attribute *amount* and dimensional characteristics *date*, *cost class*, *project*, *purchaser*, and *funding*. The values of each dimension are further arranged into hierarchies by defining the desired granularity levels, as illustrated by a diagram in ME/R notation (Multidimensional Entity/Relationship, introduced in [12]) shown in Figure 2.

We proceed by specifying various relationships within the dimensions of our case study and the requirements for their modeling.

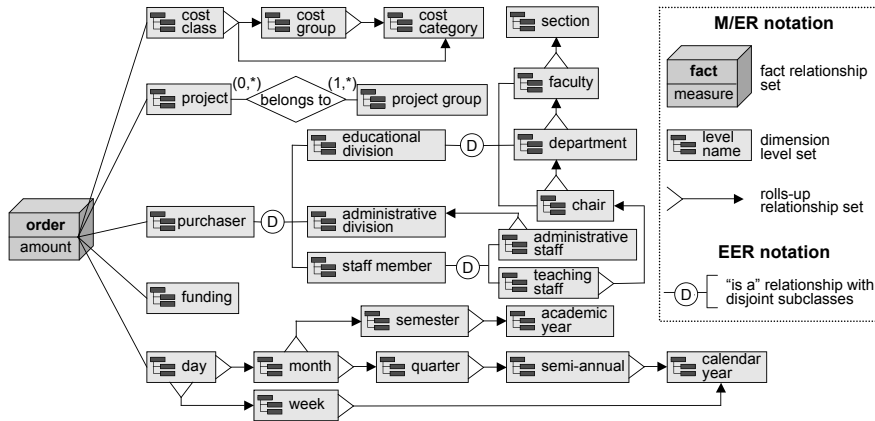


Fig. 2. University expenditures case study as ME/R Diagram

1. *Non-hierarchy*: A dimension with a single granularity, i.e. not involved in any incoming or outgoing rolls-up relationship, as is the case with *funding*.
2. *Strict hierarchy*: A dimension with only one outgoing rolls-up relationship per entity, i.e. with a many-to-one relationship towards each upper level of aggregation, for instance, *chair* → *department* → *faculty* → *section*.
3. *Non-strict hierarchy*: A dimension allows many-to-many relationships between its levels. In our example, the relationship between *project* and *project group* allows a single project to be associated with multiple project groups.
4. *Multiple hierarchies*: A single dimension may have several aggregation paths, as in *period*, where *day* may be grouped by *month* → *quarter* → *semi-annual* → *calendar year*, or by *week* → *calendar year*, or by *month* → *semester* → *academic year*. The former two paths are called *alternative* since they aggregate to the same top level.
5. *Heterogeneous hierarchy*: Consider the *purchaser* entity which is a super-class of *educational division*, *administrative division*, and *staff member*. Each sub-class has its own attributes and aggregation levels resulting in heterogeneous subtrees in the data hierarchy. Another example is *staff member* with sub-division into *administrative staff* and *teaching staff*.
6. *Non-covering hierarchy*: Strict hierarchy whose data tree is ragged due to allowing the links between data nodes to “skip” one or more levels. In terms of the ME/R diagram, such behavior occurs whenever the outgoing rolls-up relationship has more than one destinations level, as in *cost class*.
7. *Non-onto hierarchy*: Strict hierarchy that allows childless non-bottom nodes. For example, in the rolls-up relationship *administrative staff* → *administrative division* a division may appear to have no staff in purchaser role.
8. *Mixed-granularity hierarchy*: The data tree is unbalanced due to mixed granularity, as in the case of *educational division* whose sub-classes are, on the one hand, the end-instances of *purchaser* dimension, but, on the other hand, serve as aggregation levels in the hierarchy *chair* → *department* → *faculty*.

3 Extending the Multidimensional Data Model

In our work we rely on the terminology and formalization introduced by Petersen et al. in [9] since their model is the most powerful w.r.t. handling complex dimensional patterns like the ones identified in the previous section. However, we have also adopted some elements of the SQL(\mathcal{H}) model [4] to enable heterogeneous hierarchies.

3.1 Basic Definitions

Intuitively, data hierarchy is a tree with each node being a tuple over a set of attributes. A dimensional hierarchy is based on a hierarchical attribute (the one directly referenced in the fact table), propagated to all levels of the tree.

Definition 3.1. A *hierarchical domain* is a non-empty set V_H with the only defined predicates $=$ (identity), \sqsubseteq (child/parent relationship), and \sqsubseteq^* (transitive closure, or descendant/ancestor relationship) such that the graph G_{\sqsubseteq} over the nodes $\{e_i\}$ of V_H is a tree. Attribute A of V_H is called a *hierarchical attribute*.

A hierarchy H is non-strict whenever $\exists(e_1, e_2, e_3 \in V_H) \wedge e_1 \sqsubseteq e_2 \wedge e_1 \sqsubseteq e_3 \wedge e_2 \neq e_3$, or, informally, if any node is allowed to have more than one parent.

Definition 3.2. A *hierarchy schema* \mathcal{H} is a four-tuple $(\mathcal{C}, \sqsubseteq_{\mathcal{H}}, \top_{\mathcal{H}}, \perp_{\mathcal{H}})$, where $\mathcal{C} = \{\mathcal{C}_j, j = 1, \dots, k\}$ are category types of \mathcal{H} , $\sqsubseteq_{\mathcal{H}}$ is a partial order on the \mathcal{C}_j 's, and $\top_{\mathcal{H}}$ and $\perp_{\mathcal{H}}$ are the top and bottom levels of the ordering, respectively.

\mathcal{C}_j is said to be a category type in \mathcal{H} , denoted $\mathcal{C}_j \in \mathcal{H}$, if $\mathcal{C}_j \in \mathcal{C}$. Predicates \sqsubseteq and \sqsubseteq^* are used to define child/parent and descendant/ancestor relationship, respectively, between the category types in \mathcal{C} .

Definition 3.3. A *hierarchy (instance)* H associated with hierarchy schema \mathcal{H} is a two-tuple (C, \sqsubseteq) , where $C = \{\mathcal{C}_j\}$ is a set of categories such that $Type(\mathcal{C}_j) = \mathcal{C}_j$ and \sqsubseteq is a partial order on $\cup_j \mathcal{C}_j$, the union of all dimensional values in the individual categories.

A category \mathcal{C}_j is a set of dimensional values e such that $Type(e) = \mathcal{C}_j$; $|\mathcal{C}_j|$ returns the number of values in set \mathcal{C}_j . Hierarchy's data is stored in collection of tables with at most one table per schema node. Unlike in the original model of Jagadish et al. [4], we do not disallow tables with straddling levels in order to enable modeling of non-covering and mixed-granularity hierarchies.

We are now ready to formalize the notion of a homogeneous dimension.

Definition 3.4. A *homogeneous dimension* \dot{D} is defined by its hierarchy schema $\mathcal{H} = (\mathcal{C}, \sqsubseteq_{\mathcal{H}}, \top_{\mathcal{H}}, \perp_{\mathcal{H}})$ and the respective hierarchy instance $H = (C, \sqsubseteq)$.

$\perp_{\mathcal{H}}$ is the type of $\hat{\mathcal{D}}$'s bottom category, i.e. the one containing the values of the finest granularity; $\top_{\mathcal{H}}$ corresponds to an abstract root node with a single value \top , also referred to as *ALL*.

A heterogeneous dimension is defined as consisting of multiple sub-dimensions, unified into a single hierarchy by means of super-classing:

Definition 3.5. A *heterogeneous dimension* $\hat{\mathcal{D}}$ is a pair $(\mathcal{D}, \top_{\mathcal{D}})$ where $\mathcal{D} = \{\mathcal{D}_i\}$ is a set of sub-dimensions and $\top_{\mathcal{D}}$ is an abstract super-class root node. Each sub-dimension \mathcal{D}_i is of type $\hat{\mathcal{D}}$ or $\hat{\mathcal{D}}$.

Figure 3 shows the resulting dimensional fact schema of our case study.

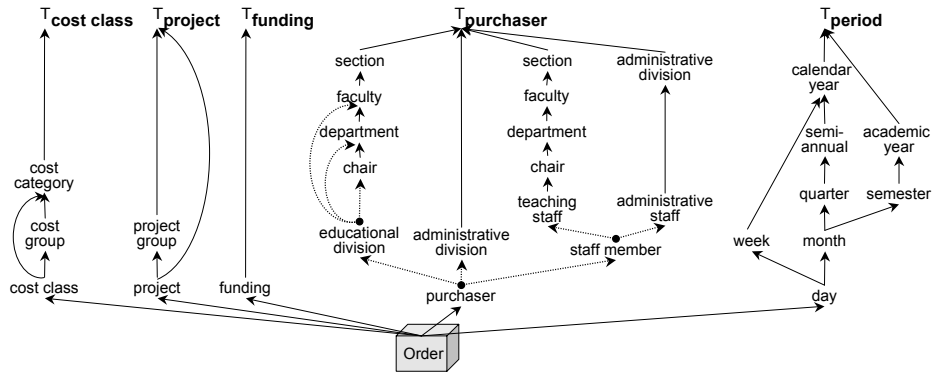


Fig. 3. University expenditures cube as 5-dimensional fact schema

3.2 Modeling Heterogeneous Hierarchies

At the conceptual level, heterogeneity corresponds to an *is_a* relationship, i.e. where the instances of a super-class are divided into sub-classes, each with its own attributes and aggregation levels. Logically, a super-class corresponds to an upper aggregation level w.r.t. its sub-class categories, but in the M/ER model super-classing is used for “homogenizing” heterogeneous entities and thus, a super-class ends up being a child of its sub-classes. Back to Figure 3, notice that super-classes *purchaser*, *educational division*, and *staff member* had to be placed underneath their respective sub-classes in the hierarchical schema.

From the logic of aggregation, the position of super-class entities is an obvious misplacement provoked by the requirement to have a single bottom granularity per dimension, so that it can be referenced by one foreign key in the fact table.

In Figure 4 we show the dimensional hierarchy of *purchaser* obtained by following the logic of dis-aggregation¹. Notice how the heterogeneity of the dimensional

¹ Attached to each category node is the number of dimensional bottom-level values covered by that category. Unlike standard hierarchical categories, a sub-class of an *is_a* relationship contains just a fraction of its parent’s values.

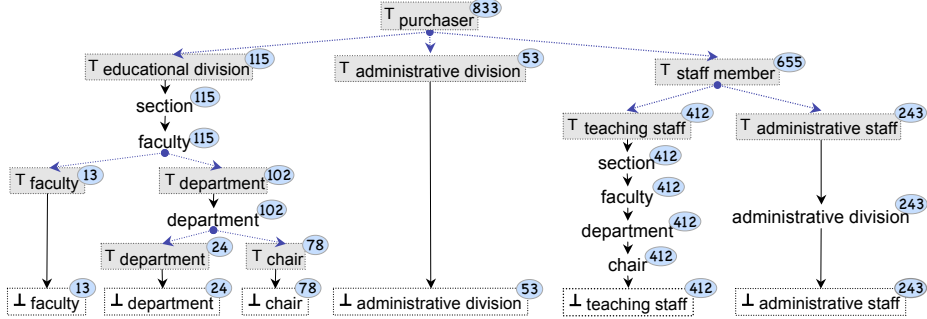


Fig. 4. Reshaping heterogeneous dimensions using abstract nodes

data has become obvious even at the bottom level. Using a straightforward intuition about hierarchically decomposing an aggregate, we can now derive a rule for modeling a *heterogeneous hierarchy*:

- ▷ the most general super-class serves as the root category $\top_{\mathcal{D}}$ whereas any further super-classes are normal categories;
- ▷ sub-classes are multiple child categories of their super-class category;
- ▷ sub-class category is of abstract type $\top_{\mathcal{D}_i}$ since it plays the role of an abstract root node for sub-dimension \mathcal{D}_i ;
- ▷ sub-class entity is used repeatedly as a non-abstract bottom category $\perp_{\mathcal{D}_i}$ if it corresponds to the finest granularity of \mathcal{D}_i .

3.3 Modeling Mixed-Granularity Hierarchies

A special case of heterogeneity is a mixed-granularity hierarchy in which sub-classes of an *is_a* relationship are also used as hierarchy levels, as observed in educational institution where *faculty* and *department* are purchasers in their own right and also serve as aggregation categories for *chair*.

Our approach to modeling mixed-granularity is a straightforward mapping of the two-fold nature of its categories by means of sub-classing: mixed-granularity category is viewed as a heterogeneous dimension sub-divided into a non-hierarchical and a hierarchical sub-dimension, corresponding to its respective two roles. Further, the general rule of heterogeneous dimension modeling is applied. The resulting schema for educational division is shown in Figure 4.

4 Schema-Based Navigational Framework

Analysts interact with OLAP data in a predominantly “drill-down” fashion, starting with highly aggregated values and descending step-wise to the desired dimensionality and level of detail. The analyst’s task can be thus reduced to a) selecting the measure and the aggregation function, b) browsing to the desired granularity in dimensional hierarchies, and c) filtering data to define the subset

to display. The visual OLAP interface is divided into two major areas of interaction: a navigation panel for browsing through dimensional data and the main window for displaying query results. Selection of measures, functions, dimensional levels and values is done using the mouse, by clicking, marking, dragging and so on.

A fact table is represented by a top-level folder (cube icon) with sub-divisions DIMENSIONS and MEASURES. Each hierarchical dimension is a folder containing its schema categories as nested subfolders, from the root category \top at the top-level to the bottom category \perp , the latter represented by a page icon. Non-abstract categories are supplied with a button for displaying their actual data. Figure 5 shows the navigational structure of our case study’s OLAP cube.

In the remaining subsections we present the techniques for mapping all types of dimensional hierarchies described in section 2 to a schema-based navigational hierarchy.

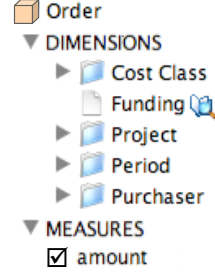


Fig. 5. Fact table navigation

4.1 Hierarchy Normalization Techniques

Schema-based navigation works correctly, if each data instance strictly adheres to the schema of its respective hierarchy, or, formally, if for any two categories C_j, C_i such that $C_i \sqsubset C_j$ the following summarizability conditions hold:

1. The mapping is *covering*: $\forall e_1 \in C_i : \exists e_2 \in C_j \wedge e_1 \sqsubseteq e_2$,
2. The mapping is *onto*²: $\forall e_2 \in C_j : (\exists e_1 \in C_i \vee (\exists e_1 \in C_k \wedge C_k \sqsubset C_j)) \wedge e_1 \sqsubseteq e_2$,
3. The mapping is *strict*: $\forall e_1 \in C_i : e_2, e_3 \in C_j \wedge e_1 \sqsubseteq e_2 \wedge e_1 \sqsubseteq e_3 \Rightarrow e_2 = e_3$.

Handling of non-summarizable data depends largely on the semantics behind that data. If irregularity is caused by missing or imprecisely captured values and it is crucial to produce imprecision-aware queries and results (e.g., in clinical diagnosing or risk assessment), the approach of Pedersen et al. [9], in which the original data remains un-normalized and imprecision is made explicit to the user by providing a set of alternative queries, may be an appropriate solution.

However, if the data hierarchy is intrinsically irregular, as is **project** dimension, where a project may be assigned to multiple groups or not assigned to any, such data should be normalized to become navigable in a uniform way.

We adopt and modify the dimension transformation technique proposed by Pedersen et al. in [7]. The original algorithm normalizes irregular hierarchies by enforcing the summarizability conditions in the above order. The whole 3-step transformation process, exemplified by normalizing the **project** dimension is shown in Figure 6. In the second step, we provide options b) and c) in addition to the original option a). *Onto* is enforced in the last step and can be omitted altogether since missing bottom-level values are not relevant for navigability.

² By considering another child C_k we account for contingent heterogeneity of C_j .

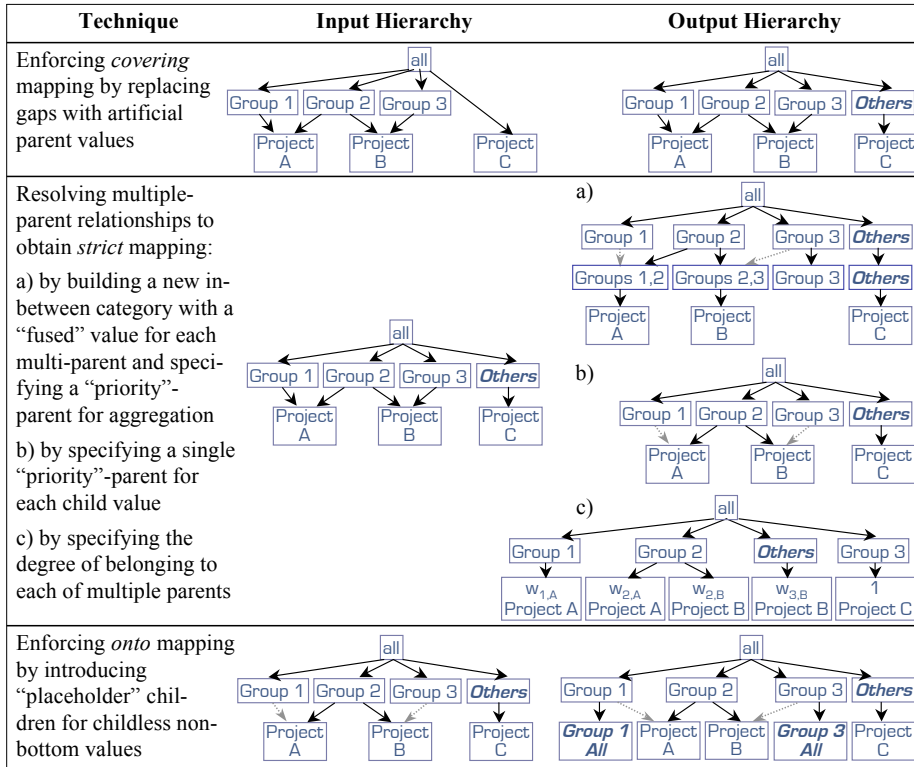


Fig. 6. 3-step normalization of the irregular dimension *project*

4.2 Schema Transformation Techniques

The navigational structure of a dimension is a recursive nesting of sub-dimensional nodes, where each node is used for drilling down to the respective granularity. The results of a drill-down are the sub-aggregates computed for each dimensional value. With respect to its underlying data hierarchy, the behavior of a sub-dimensional schema node can be reduced to the following types:

- ▷ *Non-hierarchical*, i.e bottom level, displayed as a non-expandable page icon;
- ▷ *Single-hierarchy* node is a folder containing a single subfolder of its child;
- ▷ *Multiple hierarchy* contains a subfolder for each of the alternative paths. These paths are mutually exclusive, so that once the user has selected one path, all others should be disabled for further interaction;
- ▷ *Super-class* is a folder containing all sub-class categories as subfolders. Since the super-class has no data of its own, there is no data display option. However, drill-down is possible and produces the aggregates of the sub-class categories. Sub-class folders are visually linked to each other, to be distinguished from the multiple hierarchy case since the former are not exclusive and, therefore, can be further explored in parallel;

- ▷ *Abstract Root*, node is a top-level folder with no data, used purely as a “wrapper” for the entire dimensional schema nested therein. Notice that abstract root is superfluous in case of a non-hierarchical (nothing to “wrap”) or heterogeneous (abstract root already available) dimension.
- ▷ *Mixed-granularity* is a complex hierarchical node subdivided into a hierarchical and a bottom-level sub-dimensions.

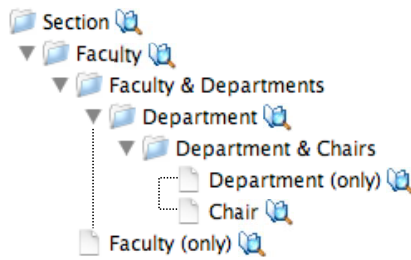


Fig. 7. Schema navigation hierarchy for a mixed-granularity fragment

Mixed-granularity deserves special attention due to its complexity. Figure 7 shows the resulting navigation for the fragment $\text{section} \rightarrow \text{faculty} \rightarrow \text{department} \rightarrow \text{chair}$. Its structure is derived from the schema depicted in Figure 4, with the exception that the artificial sub-classes, such as \top_{faculty} and $\top_{\text{department}}$ are merged into a common superclass node **Faculty & Departments**. This node is abstract and thus behaves as expected, i.e., its drill-down displays each of the two

sub-class aggregates. The resulting navigation structure is rather complex, but it enables retrieval of a wide spectrum of aggregates with mere “drag-and-drop” interactions.

We have implemented the presented schema-based exploration approach for complex OLAP data as a Java application which connects to a specified database and allows user to navigate in OLAP cubes presenting the results as a pivot table, chart or a decomposition tree. At this stage, performance and scalability issues were left out of consideration.

5 Conclusion and Future Work

Inspired by the growing demand for OLAP applications in novel domains, confronted with irregular multidimensional data, we have presented a framework for modeling complex hierarchical dimensions and their seamless mapping to a schema-based navigational structure of a visual OLAP interface. Using a case study from the area of academic administration, we have provided a classification of dimensional behaviors, leading to non-summarizable hierarchies, such as ragged, unbalanced or non-strict data trees, as well as heterogeneous or mixed-granularity dimensional schema.

Our approach is based on a two-phase transformation of irregular dimensions: 1) enforcing summarizability within single homogeneous data hierarchies, and 2) reshaping complex hierarchical schemata into a set of well-behaved sub-dimensions. Our model does not introduce any query language extensions; it rather relies on the meta-data (e.g., dimension type, hierarchy schema, category type) for mapping OLAP data to a visual browser and translating user interaction back to the database operations.

Among our future research directions are to provide explicit handling of temporal and spatial aspects in modeling and querying OLAP data, to investigate the applicability of schema-based browsing for semi-structured and high-dimensional data, and to search for novel visualization and interaction techniques capable of presenting large volumes of complex data for explorative analysis.

References

1. “Cognos PowerPlay: Overview—OLAP Software,” 2006. [Online]. Available: <http://www.cognos.com/powerplay>
2. E. F. Codd, S. B. Codd, and C. T. Salley, “Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate,” *Technical report, E.F.Codd & Associates*, 1993.
3. C. A. Hurtado and A. O. Mendelzon, “Reasoning about summarizability in heterogeneous multidimensional schemas,” in *ICDT 2001, Proceedings of the 8th International Conference on Database Theory*, 2001, pp. 375–389.
4. H. V. Jagadish, L. V. S. Lakshmanan, and D. Srivastava, “What can hierarchies do for data warehouses?” in *VLDB ’99, Proceedings of 25th International Conference on Very Large Data Bases*, 1999, pp. 530–541.
5. H.-J. Lenz and A. Shoshani, “Summarizability in OLAP and statistical data bases,” in *Proceedings of 9th International Conference on Scientific and Statistical Database Management*, 1997, pp. 132–143.
6. T. Niemi, J. Nummenmaa, and P. Thanisch, “Logical multidimensional database design for ragged and unbalanced aggregation,” in *Proceedings of 3rd International Workshop on Design and Management of Data Warehouses*, 2001, pp. 7.1–7.8.
7. T. B. Pedersen, C. S. Jensen, and C. E. Dyreson, “Extending practical pre-aggregation in on-line analytical processing,” in *VLDB’99, Proceedings of 25th International Conference on Very Large Data Bases*, 1999, pp. 663–674.
8. —, “The TreeScape system: Reuse of pre-computed aggregates over irregular OLAP hierarchies,” in *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases*, 2000.
9. —, “A foundation for capturing and querying complex multidimensional data,” *Information Systems*, vol. 26, no. 5, pp. 383–423, 2001.
10. M. Rafanelli and A. Shoshani, “STORM: A statistical object representation model,” in *Proceedings of 5th International Conference on Statistical and Scientific Database Management*, 1990, pp. 14–29.
11. “SAP NetWeaver Business Intelligence,” 2006. [Online]. Available: <http://www.sap.com/solutions/netweaver/components/bi>
12. C. Sapia, M. Blaschka, G. Höfling, and B. Dinter, “Extending the E/R model for the multidimensional paradigm,” in *ER ’98, Proceedings of the Workshops on Data Warehousing and Data Mining*, 1999, pp. 105–116.
13. “Tableau software,” 2006. [Online]. Available: <http://www.tableausoftware.com>
14. S. Vinnik and F. Mansmann, “From analysis to interactive exploration: Building visual hierarchies from OLAP cubes,” in *EDBT 2006, Proceedings of 10th International Conference on Extending Database Technology*, 2006, pp. 496–514.
15. T. Zurek and M. Sinnwell, “Datawarehousing has more colours than just black & white,” in *VLDB ’99, Proceedings of 25th International Conference on Very Large Data Bases*, 1999, pp. 726–729.