VISUAL OLAP: A NEW PARADIGM FOR EXPLORING MULTIDIMENSIONAL AGGREGATES

Svetlana Mansmann and Marc H. Scholl
University of Konstanz
Box D188, 78457 Konstanz, Germany

ABSTRACT

OLAP (On-Line Analytical Processing) technology provides interactive query-driven analysis of accumulated and consolidated business data for the purpose of decision-making and knowledge extraction. Visualization is increasingly used as the means of gaining insight into huge data volumes of multidimensional data as the former exploits the profound ability of the human vision system to recognize patterns. This paper describes a new generation of analysis tools, characterized as *Visual OLAP* and based on the emerging visual exploration paradigm. In contrast to conventional tools that employ visualization primarily as user-friendly data presentation, visual OLAP turns visualization into the key method for both query specification and exploratory analysis. We propose a comprehensive visual exploration framework, which implements OLAP operations in form of powerful data navigation and allows users to explore data using a variety of interactive visualization techniques. A taxonomy of visual metaphors, from classical business charts to advanced 3D, multiscale, and hierarchical layouts accounts for different types of analytical tasks. A unified framework is obtained by abstracting various visualization options into a common presentation model and providing algorithms for mapping user interactions to database queries as well as mapping query results to a specified visual layout.

KEYWORDS

Visual OLAP, Information Visualization

1. INTRODUCTION

OLAP (On-Line Analytical Processing) technology, a term coined by Codd (1993), provides interactive query-driven analysis of accumulated and consolidated business data for the purpose of decision making and knowledge extraction. The first generation of OLAP frontend tools was aimed primarily at satisfying the needs of routine reporting. Those tools provide a *managed query environment* that limits the end users to navigating within a set of pre-defined queries [Chaudhuri, S. and Dayal, U., 1997]. However, with the recent achievements in the information technology, the scope of supported analytical tasks has expanded far beyond interactive report generations. Comprehensive analysis includes a variety of task types such as examining the data from multiple perspectives, extracting useful information, verifying hypotheses, recognizing trends, revealing patterns, and discovering new knowledge from arbitrarily large and complex data volumes.

Visual OLAP approach overcomes the limitations of conventional interfaces by unlocking the synergy between the performance-oriented *Business Intelligence* techniques and the achievements in the areas of *Information Visualization*, *Human-Computer-Interaction*, and *Visual Analytics*. Business intelligence is an umbrella term encompassing a broad category of applications and technologies for gathering, storing, analyzing, and providing access to data to facilitate decision-making and improve business performance. To efficiently analyze huge data volumes and uncover the "hidden gems" therein, novel tools enable freewheeling data exploration allowing users to navigate to the desired view, experiment with various layouts, thus supporting the process of incrementally refining a question into an answer. Figure 1 shows the data exploration process, also denoted *knowledge crystallization*, adopted from [Card, S.K. et al., 1999] and slightly modified for the context of OLAP. In this cycle, visualization clearly plays the key role in providing insight into the data and, thus, solving the task. Visualization has the power to save time and reduce errors in analytical reasoning by utilizing the phenomenal abilities of the human vision to recognize patterns [Hanrahan, P. et al., 2007].

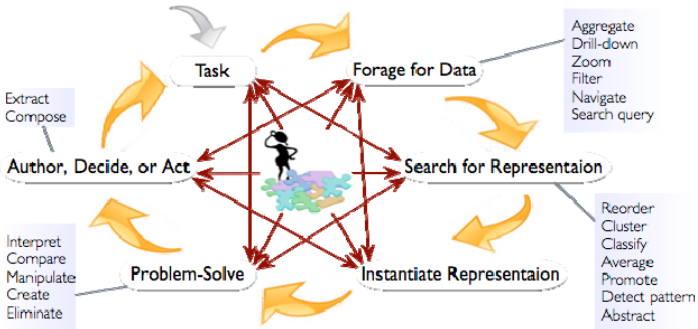


Figure 1. Data exploration cycle.

This paper is structured as follows: Section 2 introduces the fundamental concepts of OLAP, followed by the related work in Section 3. Section 4 describes the process of visual query specification. A categorization of visualization techniques is given in Section 5, concluded by the summary of our contribution in Section 6.

2. STATE-OF-THE-ART OLAP TOOLS AND RELATED WORK

A traditional interface for analyzing OLAP data is a *pivot table*, or *cross-tab*, which is a multidimensional spreadsheet produced by specifying one or more measures of interest and selecting dimensions to serve as vertical (and, optionally, horizontal) axes for summarizing the measures. The power of this format comes from its ability to summarize detailed data along various dimensions and arrange the aggregates computed at different granularity levels into a single view preserving the "part-of" relationships between the aggregates. Figure 2 (right) exemplifies the idea of "unfolding" a 3-dimensional data cube into a pivot table.

Pivot tables are commonly criticized for disgraceful handling of large data sets and inefficiency for solving non-trivial analytical tasks, such as recognizing patterns, discovering trends, identifying outliers, etc. State-of-the-art OLAP interfaces enhance the pivot table view by providing a wealth business visualization techniques, from popular business charts to more exotic and sophisticated layouts as well as vendors' proprietary visualizations. Some tools go beyond mere visual presentation of data and propose sophisticated approaches inspired by the findings in information visualization research.

Prominent examples of advanced visual systems are Advizor [Eick, S.G., 2000] and Tableau [Hanrahan, P. et al., 2007]. The Advizor's technique organizes data into three perspectives. A perspective is a set of linked visual components displayed together on the same screen. Each perspective focuses on a particular type of analytical task, such as 1) single measure view using a 3D multiscape layout, 2) multiple measures arranged into a scatterplot, and 3) anchored measures presented using techniques from multidimensional visualization (box plots, parallel coordinates, etc.). Tableau is a commercialized successor of Polaris - a visual tool for multidimensional analysis developed at Stanford University [Stolte, C. et al., 2002]. Polaris inherits the basic idea of the classical pivot table interface, however, it uses embedded graphical marks rather than textual numbers in the table cells. The types of supported graphics are arranged into taxonomy.

While most vendors tend to limit the scope of supported visual layouts to popular and proven ones, researchers propose to employ novel visualization techniques to take full advantage of multidimensional and hierarchical properties of the data. Tegarden (1999) formulates the general requirements of business information visualization and gives an overview of advanced visual metaphors for multivariate data, focusing on 3D techniques, such as *3D Scattergrams*, *3D line graphs*, *floors and walls*, and *3D map-based bar charts*.

Another branch of visualization research for OLAP concentrates on developing *multiscale* visualization techniques capable of presenting the data at different levels of aggregation. [Stolte, C. et al., 2002] describe their implementation of multiscale visualizations within the framework of the Polaris system. The underlying visual abstraction is that of a *zoom graph* that supports multiple zooming paths, where zooming actions may be tied to dimension axes or triggered by a different type of interaction.

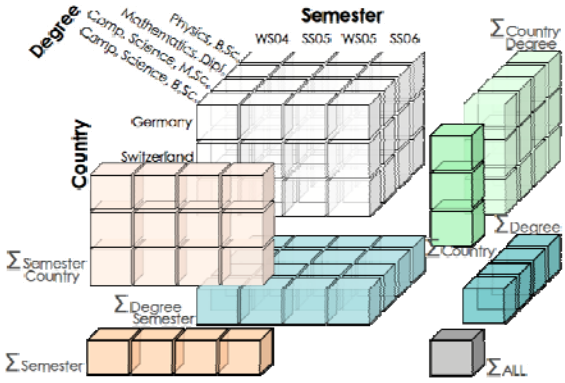
[Lee, H.-Y. and Ong, H.-L., 1997] proposed a multidimensional visualization technique that adopts and modifies the *Parallel Coordinates* method for knowledge discovery in OLAP. The main advantage of this technique is its scalability to virtually any number of dimensions. Each dimension is represented by a vertical

axis and the aggregates are aligned along each axis in form of a bar chart. The other side of the axis may be used for generating a bar chart at a higher level of detail.

[Mansmann, S. and Scholl, M.H., 2007] concentrate on the problem of losing the aggregates computed at preceding query steps while changing the level of detail and propose to use hierarchical layouts for capturing the results of multiple decompositions within the same display using the *Enhanced Decomposition Tree* technique. An advanced exploration framework for OLAP based on *coordinated views* of dimension hierarchies is proposed in [Sifer, M., 2003]. Each dimension hierarchy with qualifying fact entries attached as the bottom-level nodes is presented using a space-filling nested tree layout. Drilling-down and rolling-up is performed implicitly by zooming within each dimension view. An interactive visualization technique, called *Hierarchical Dynamic Dimensional Visualization*, is proposed in [Techapichetvanich, K. and Datta, A., 2005]. Dimension instances are shown as hierarchically aligned barsticks partitioned into rectangles that represent portions of the aggregated measure associated with the respective dimension member. Color intensity indicates the density of the number of records satisfying a specified range condition.

3. OLAP FUNDAMENTALS

OLAP technology draws its analytical power from the underlying *multidimensional data model*. The data is modeled as cubes of uniformly structured *facts*, consisting of analytical values, referred to as *measures*, uniquely determined by descriptive values drawn from a set of *dimensions*. Each dimension forms an axis of a cube, with dimension members as coordinates of the cube cells storing the respective measure values. Figure 2 (left) shows a strongly simplified example of a 3-dimensional data cube, storing student enrollment numbers as a measure determined by dimensions `Country`, `Degree`, and `Semester`.



		Measures									
		• NumPersons (AVERAGE)					• NumPersons(SUM)				
Dimensions		Semester									
Country	Degree	WS04	SS06	WS05	SS06	Total	WS04	SS06	WS06	SS06	Total
– Germany	Physics, B.Sc.	114	47	120	39	80	114	47	120	39	320
	Mathematics, Dipl.	51	12	44	13	30	51	12	44	13	120
	Comp. Science, M.Sc.	21	6	18	5	12.5	21	6	18	5	50
	Comp. Science, B.Sc.	65	0	73	0	34.5	65	0	73	0	138
Total Germany		62.75	16.25	63.75	14.25	39.25	251	65	255	57	628
– Switzerland	Physics, B.Sc.	8	2	3	0	3.25	8	2	3	0	13
	4	8	9	2	1	1	1	0	2	1	4
	Comp. Science, M.Sc.	3	0	4	1	2	3	0	4	1	8
	Comp. Science, B.Sc.	5	0	4	0	2.25	5	0	4	0	9
Total Switzerland		4.25	0.5	3.25	0.5	2.125	17	2	13	2	34
– U.S.A.	Physics, B.Sc.	2	0	5	1	2	2	0	5	1	8
	3	1	2	2	0	0.75	0	1	2	0	3
	Comp. Science, M.Sc.	0	0	0	1	0.25	0	0	0	1	1
	Comp. Science, B.Sc.	2	0	0	0	0.5	2	0	0	0	2
Total U.S.A.		1	0.25	1.75	0.6	0.875	4	1	7	2	14
Total Product		22.67	5.67	22.91	5.08	14.08	272	68	275	61	676

Figure 2. A sample 3-dimensional cube for storing student enrollment numbers (left) and a pivot table with average and total student enrollment numbers grouped vertically by Country and Degree and horizontally by Semester (right)

Member values within a dimension are further organized into *classification hierarchies* to support additional aggregation levels. The levels of the hierarchy form the *dimension schema* while the hierarchy of dimension values are referred to as *dimension instance*. The attribute upon which the hierarchy is defined is called the *analysis criterion*. Multiple hierarchies may be defined within a dimension based on the same or to different analysis criteria. Hierarchies defined upon the same criterion are called *multiple alternatives*, with time dimension as a classical example, as dates maybe summarized either by week or by month, but not both. Hierarchies based on various criteria are called *parallel*, with a corresponding example of Degree dimension depicted in Figure 3: one classification is based on the attribute Degree Type while the other draws upon the attribute Subject. In contrast to multiple alternatives, parallel hierarchies may be explored in parallel. In addition to the analysis criteria attribute(s), levels in a hierarchy may include non-hierarchical characteristics denoted *properties*. In the example of Degree hierarchy, Department level may have such properties as Dean, Location, and Foundation Date.

In relational OLAP systems data cubes are stored in relations of types *fact table* and *dimension table*. A fact table stores the fact entries and is composed of two types of columns - measures and dimensions - where each dimension column is a foreign key to the respective dimension table. The primary key of a fact table is

usually a composite key made up of all its foreign keys. A dimension table is used for storing the members of each dimension along with its classification hierarchies.

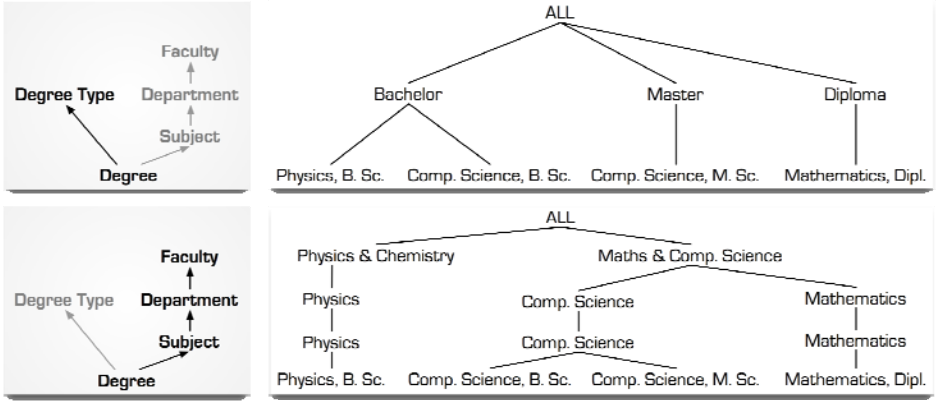


Figure 3. Dimension `Degree` with multiple hierarchies: schema (left) and instances (right)

The two logical design options are *star* schema and *snowflake* schema [Kimball, R., 1996], differing solely in the way they handle dimension hierarchies. Star schema, used in most data warehouses, places each dimension with all its hierarchies into exactly one de-normalized relation to facilitate navigation and improve query performance. Snowflake schema is a refinement of the star schema, in which each dimension hierarchy is decomposed into multiple tables, one per level, to avoid redundancy. Normalized storage is also advantageous for explicit sharing of dimensions and their parts among multiple data cubes. Multiple fact tables related via dimension sharing form a *galaxy*. Galaxy schema is very flexible and powerful, however, it comes at the expense of high design overhead because many variants of aggregation must be considered. Figure 4 shows an example of a galaxy constructed from the snowflake schemas of cubes `ENROLLMENTS` and `EXPENDITURES`. Even though the cubes do not fully share any dimension, they do have shared dimension levels `Semester` and `Department`.

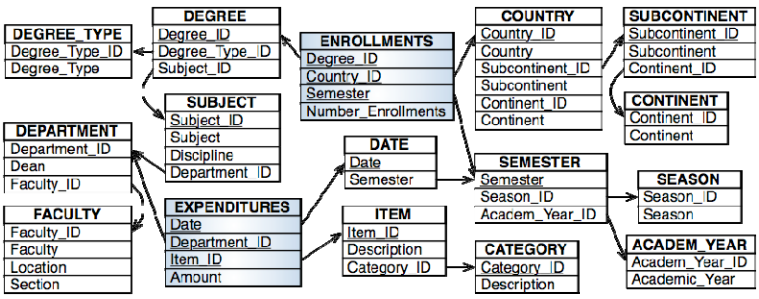


Figure 4. Galaxy schema of two fact tables with normalized dimension tables

4. VISUAL QUERY SPECIFICATION

Visual OLAP disburdens end-users from composing queries in the "raw" database syntax (e.g., SQL or MDX). Instead, queries are specified visually. Multidimensional data is represented as a browsable structure that can be explored interactively. Visual interface does not trade off advanced functionality for simplicity, but rather facilitates the process of specifying ad hoc queries of arbitrary complexity.

A common data-browsing paradigm is that of a navigation tree, i.e., a recursive nesting of element nodes. The nodes in the navigation scheme may be of types *database*, *schema*, fact *table* (*cube*), *dimension*, *classification level*, and *measure*. In simplified configurations, the navigation may be limited to single data cubes and, thus, consist solely of dimension and measure attributes of a selected cube. Data cubes as navigation objects consist of measure and dimension nodes. A dimension is represented either as a lattice of its classifi-

cation levels or directly by the data hierarchy. Standard navigation schemes with direct display of dimensional data trees fail to handle complex dimensions, such as non-strict, heterogeneous or unbalanced mappings, as well as parallel and multiple alternative hierarchies. Besides, they cannot support advanced query options, such as joining multiple cubes or interchanging the roles of dimensions and measures.

Our proposed navigation scheme adopts the intension-based approach introduced in [Mansmann, S. and Scholl, M.H., 2006]. Figure 5 depicts the resulting navigation of cube ENROLLMENTS, with dimension hierarchies expanded to display their categories. Members of a category are displayed on demand and can also be browsed in a hierarchical fashion, as shown at the example of querying the data of Degree Type.

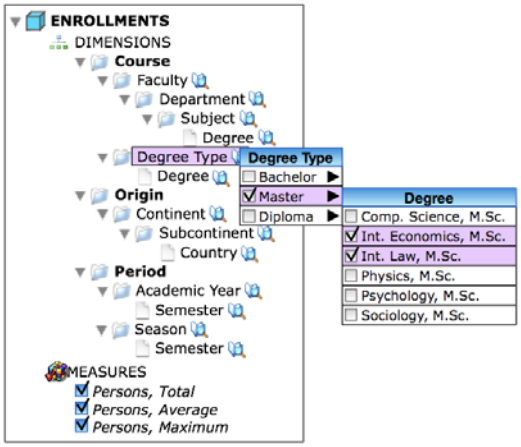


Figure 5. Intension-based cube navigation schema with on-demand display of the category's member values

Various navigation events, such as dragging and clicking, are translated into valid queries, executed instantaneously. From the user's point of view, querying is done implicitly by populating the visualization with data and incrementally refining the view. In the first step, an empty visualization template is instantiated with data by dragging the attributes of interest into the respective areas of the visualization. Any aggregation query follows the same scheme, i.e., consists of the same query clauses, some of which are optional. In SQL, a query is structured as follows (optional clauses and elements are placed in square brackets):

```
SELECT [ dimension attribut, ] measure list
FROM table list
[ WHERE predicate list ]
[ GROUP BY [ ROLLUP | CUBE ] dimension list ]
[ HAVING measure predicate list ]
[ ORDER BY attribute list [sort direction] ]
```

A query is generated automatically from the action of dropping a measure from the navigation into the visualization. Further clauses serve for refining the initial query: *i)* **WHERE** and **HAVING** clauses allow to specify selection conditions on any attributes and aggregated measure values, respectively, *ii)* **GROUP BY** contains the dimensions to aggregate along, and *iii)* **ORDER BY** provides sorted output. These clauses are populated with data by invoking the corresponding OLAP operations.

4.1 Frontend Implementation of OLAP Operations

OLAP operators take a data cube as an input and output a new cube. These operations are defined at logical level and have to be implemented in a visual framework in form of navigation or other interaction options. Our framework supports all standard as well as a number of extended and vendor specific operations. To account for various user preferences, most of the operators are provided redundantly, e.g., via the navigation, as a menu option, and via visual interaction. Operators can be subdivided by function into the following groups:

1) DRILL-DOWN and ROLL-UP operators along with their variants (DRILL-THROUGH, DRILL-WITHIN, DRILL-ASIDE, DRILL-ANYWHERE, PROJECT) manipulate dimensionality and granularity of the output data cube by adding/removing dimension categories in the **GROUP BY** clause.

2) SLICE&DICE and its special cases (SLICE, DICE, SELECT, FILTER, CONDITIONAL HIGHLIGHTING) reduce the size of the data set by adding filtering conditions to the **WHERE** clause.
3) RANKING operator applies the filtering condition on the aggregated values themselves (i.e., via the **HAVING** clause) and sorts the qualifying data entries according to their ranking.
4) PIVOT and SWITCH enable visual reordering of the output without any changes in the data set itself.
5) DRILL-TROUGH, PUSH and PULL allow to manipulate the very structure of the data cubes as to define new measures and dimensions and combine data from multiple cubes.

Table 1 shows a selection of major supported operators and their implementation options as navigation events and as interaction techniques.

Table 1. Extended set of OLAP operations with their implementation options

Operation	Description	Navigation	Interaction
ROLL-UP	decreases granularity within a dimension		*Zoom-out* within the respective dimension axis or *delete* a category from its visual mapping
PROJECT	decreases dimensionality by aggregating across the entire dimension		*Delete* a dimension from the visualization
DRILL-DOWN	deepens granularity within a dimension	*Dragging* a category node into the visualization	*Zoom-in* within the respective dimension axis or *expand* the element of interest
DRILL-THROUGH	retrieves the actual fact data behind the aggregates		Corresponding menu *button* or a *popup menu* option
DRILL-WITHIN	drills into another hierarchy of the same dimension	s. DRILL-DOWN	A *popup menu* option
DRILL-ANYWHERE	drills into a dimension outside of the previous path	s. DRILL-DOWN	A *popup menu* option
SLICE	reduces dimensionality by filtering a dimension in the drill path to a single value	*Selecting* a single value in a category to serve as a filter	*Trim* the view to the area of interest
DICE	specifies values to be excluded from a dimension in the drill path	*Selecting* values in a category to be filtered out	*Deleting* or *collapsing* corresponding areas in the visualization
SELECT	reduces a dimension in the drill path to a set of values or to a certain value range	*Selecting* values in a category to serve as a filter	*Trim* the view to the area of interest; *sliders* or *panning window* may be used for range selections
SLICE&DICE	selects a sub-cube by combining slicing and dicing	*Selecting* values in multiple dimensions as a filter	Iteratively *trim* the view to the area of interest
FILTER	specifies selection conditions on dimensions not in the drill path, thus affecting the aggregated values	s. DICE	Configuration via a *filter* menu or by using *sliders* for range selection
RANKING	outputs the top/bottom *n* cube cells with respect to the ranking function	Editing the measure's definition to include the ranking function	Layout-specific options, a *filter* menu or a *slider* on the measure value
SWITCH	re-orders the aggregates in the visualization		*Sort* menu or *dragging* visual elements in into new positions
PIVOT	changes dimensional orientation of the view		*Dragging* the affected data fields to their new visual mappings
DRILL-ACROSS	joins multiple related data cubes to combine their measures	Switching to a *multi-cube navigation* scheme	
PUSH	converts a dimension attribute into a measure	Measure definition wizard	
PULL	converts a measure attribute into a dimension	*Dragging* the measure into the list of dimensions	

5. VISUALIZATION TOOLKIT

In the context of OLAP, visualization refers to the mapping of the data returned by a query (or a series of queries) to a visual layout. Visual presentation is generated by assigning the cube's elements - measures and dimensions - to the visual variables of the display. A visualization technique is defined by the visual variables, or attributes, it employs and the way those variables are combined. Note that certain visual properties, such as *shading*, *depth*, and *texture*, are hardly applicable in the context of OLAP due to their limited precision. There following visual attributes encountered in the layouts used for visualizing multidimensional data:

- *Position* is the placement of an element on the display.
- *Shape*, such as rectangular, circular, etc., determines the visual form and borders of single elements. *Symbols* may be used instead of shapes.
- *Size*, such as length or area, is used to scale the elements according to numeric values they represent.
- *Color* is a powerful means of mapping both discrete and continuous value domains.
- *Orientation* is applicable to certain shape types to map an additional characteristic of an element.
- *Motion* is an intuitive way of showing the evolution of a measure's value over time.

Some attributes can even be broken down to their components to be used for mapping more than one data field. For example, the color's *hue*, *saturation* and *luminosity* can be used to code three properties, and orientation can be decomposed into *angle* and *direction*. Various attributes behave differently with respect to the range, data type, and the number of values they can meaningfully represent [Mackinlay, J., 1986].

Dimensionality of a visual format is determined by its layout, with *grid* and *hierarchy* as the most popular options. For example, simple bar and pie charts have only one dimension axis, pivot tables have two axes and support nesting of multiple dimensions along each axis, whereas 3D scattergrams show a measure grouped by three dimensions. Hybrid layouts achieve higher dimensionality by means of combining multiples views into a common display. A popular *small multiples* approach [Tufte, E.R., 1986] arranges multiple views of the same granularity into a grid, thus enabling straightforward comparison of aggregates. *Enhanced decomposition tree* technique [Mansmann, S. and Scholl, M.H., 2007] enables comparison at multiple granularity levels by arranging the views generated by a series of drill-down steps into a visual hierarchy.

Various analysis tasks require various visualization techniques. OLAP tools of the state of the art leave it up to the user to choose an appropriate. In the data warehousing research, the issue of assessing the aptitude of a visualization technique to support different types of analysis tasks and OLAP operations is also left without consideration. As a result, users often come up with inefficient and even misleading visualizations.

Figure 6 shows a simple classification of visualization techniques for OLAP into four quadrants according to layout (simple vs. hybrid) and grain (uniform vs. mixed). In each quadrant, the techniques are sorted upwards in the increasing number of dimensions they can support. Visual metaphors displaying a single measure are shown with white background. This enumeration is by no means exhaustive and lists just the major techniques already implemented by OLAP tools or proposed in the literature. Descriptions of the mentioned techniques may be found in standard literature on information visualization [Tufte, E.R., 1986; Card, S.K. et al., 1999] as well as in research publications.

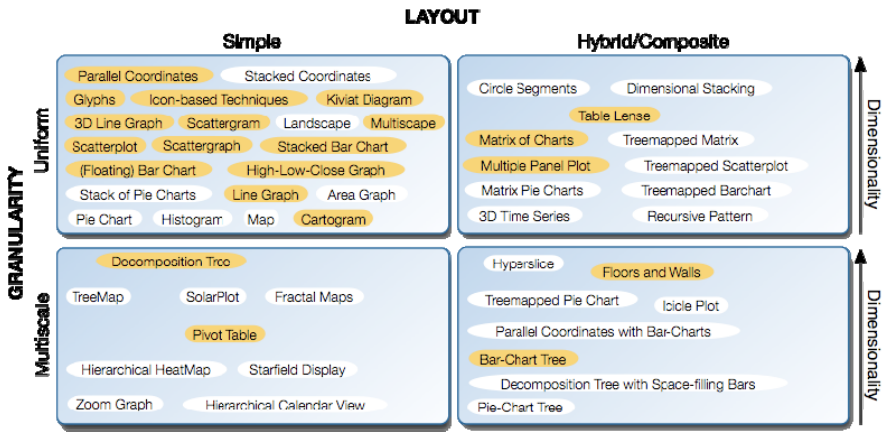


Figure 6. Segmentation of visualization techniques for OLAP by layout and granularity

Any of the state-of-the-art OLAP tools implement just a small subset of the visualizations listed in Figure 6, mostly from the upper-left quadrant of simple layouts. The emerging class of visual OLAP interfaces, however, will increasingly adopt complex layouts to support a wider spectrum of analysis tasks. With this work, we seek to animate researchers and practitioners to join their efforts in designing a comprehensive framework for *i)* incorporating new visualization and interaction techniques and *ii)* providing further useful classifications and evaluations of various visual metaphors in the context of OLAP. Eventually, the outcomes of the classification efforts may be utilized for enhancing the "intelligence" of the interface in assisting the user, e.g., in form of hints or warnings, suggesting the optimal visualization technique for a task at hand, etc.

6. CONCLUSIONS

We presented an emerging class of visual OLAP tools, in which visualization plays the key role in both presenting and exploring multidimensional cubes. These tools aim at unlocking the synergy between Business Intelligence, Information Visualization, and Visual Analytics. The overall data exploration framework consists of a powerful data navigation structure for mapping data attributes to a visualization layout. We provided an overview of OLAP operations and considered the options of their implementation in the navigation scheme as well as in form of direct interaction with the visual display. Different analytical tasks require different visual presentations. Definition of the visualization in terms of its visual variables enabled us to design a generic approach to mapping the data of the retrieved multidimensional cubes to a visual layout. As the means of facilitating the choice of an appropriate presentation, we proposed a classification of various visualization techniques, classified by layout, granularity, dimensionality, and the number of measures.

REFERENCES

Card, S.K. et al. (eds.), 1999. *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Chaudhuri, S. and Dayal, U., 1997. An overview of data warehousing and OLAP technology, *ACM SIGMOD Record* **26**(1), 65-74.

Codd, E.F. et al., 1993. *Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate* (Technical report), E.F.Codd & Associates.

Eick, S.G., 2000. Visualizing Multidimensional Data, *SIGGRAPH Computer Graphics* **34**(1), 61-67.

Hanrahan, P. et al., 2007. *Visual Analysis for Everyone: Understanding Data Exploration and Visualization* (Tableau White Paper), Tableau Software Inc., Online: http://www.tableausoftware.com/docs/Tableau_Whitepaper.pdf

Kimball, R., 1996. *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*, John Wiley & Sons, Inc., New York, NY, USA.

Lee, H.-Y. and Ong, H.-L., 1997. A new visualisation technique for knowledge discovery in OLAP, *PAKDD'97: Proc. of 1st Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pp. 279-286.

Mackinlay, J., 1986. Automating the design of graphical presentations of relational information, *ACM Transactions on Graphics* **5**(2), 110-141.

Mansmann, S. and Scholl, M.H., 2006. Extending Visual OLAP for Handling Irregular Dimensional Hierarchies, *DaWaK'06: Proc. of 8th Int. Conf. on Data Warehousing and Knowledge Discovery*, pp. 95-105.

Mansmann, S. and Scholl, M.H., 2007. Exploring OLAP Aggregates with Hierarchical Visualization Techniques, *ACM SAC 2007: Proc. of 22nd Annual ACM Symposium on Applied Computing*, pp. 1067-1073.

Sifer, M., 2003. A visual interface technique for exploring OLAP data with coordinated dimension hierarchies, *CIKM'03: Proc. of 12th Int. Conf. on Information and Knowledge Management*, pp. 532-535.

Stolte, C. et al., 2002. Query, analysis, and visualization of hierarchically structured data using Polaris. *ACM SIGKDD'02: Proceedings of 8th International Conference on Knowledge Discovery and Data Mining*, pp.112-122.

Stolte, C. et al., 2003. Multiscale visualization using data cubes. *IEEE Trans. on Vis. and Comp. Graphics* **9**(2), 176-187.

Techapichetvanich, K. and Datta, A., 2005. Interactive visualization for OLAP, *ICCSA 2005: Proceedings of the International Conference on Computational Science and its Applications (Part III)*, pp. 206-214.

Tegarden, D.P., 1999. Business information visualization, *Communications of the AIS* **1**(1), Article 4.

Tufte, E.R., 1986. *The visual display of quantitative information*, Graphics Press, Cheshire, CT, USA.