# Describing Multidimensional Data Through Highlights

(Discussion Paper)

Matteo Francia[1], Enrico Gallinucci[1], Matteo Golfarelli[1], Patrick Marcel[2], Verónika Peralta[2] and Stefano Rizzi[1]

[1]DISI, University of Bologna, Italy
[2]LIFAT, University of Tours, France

**Abstract**

The Intentional Analytics Model (IAM) is a new paradigm to couple OLAP and analytics. It relies on two ideas: (i) letting the user explore data by expressing his/her analysis intentions rather than the data (s)he needs, and (ii) returning enhanced cubes, i.e., multidimensional data annotated with knowledge insights in the form of model components (e.g., clusters). In this paper we propose a proof-of-concept for the IAM vision by delivering an end-to-end implementation of describe, one of the five intention operators introduced by IAM.

**Keywords**

OLAP, OLAM, Analytics, Multidimensional data, Data exploration

## 1. Introduction

Data warehousing and OLAP (On-Line Analytical Processing) have been progressively gaining a leading role in enabling business analyses over enterprise data since the early 90's. Recently, it has become more and more evident that the OLAP paradigm, alone, is no more sufficient since the enormous success of machine learning techniques has consistently shifted the interest of corporate users towards sophisticated analytical applications.

The *Intentional Analytics Model* (IAM) has been envisioned as a way to tightly couple OLAP and analytics [1]. IAM relies on two major cornerstones: (i) the users explore the data space by expressing their analysis *intentions* rather than by explicitly stating what data they need, and (ii) in return they receive both multidimensional data and knowledge insights in the form of annotations of interesting subsets of data. As to (i), five intention operators have been proposed, namely, describe [2], assess [3], explain, predict, and suggest. As to (ii), first-class citizens of the IAM are *enhanced cubes*, defined as multidimensional cubes coupled with *highlights*, i.e., sets of cube cells associated with interesting components of *models* automatically extracted from cubes [1]. An overview of the process is given in Figure 1.

The goal of this paper is to provide a proof-of-concept for the IAM vision by delivering an end-to-end implementation of the describe operator, which aims at describing one or more cube measures, possibly focused on one or more level members.

*SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy*

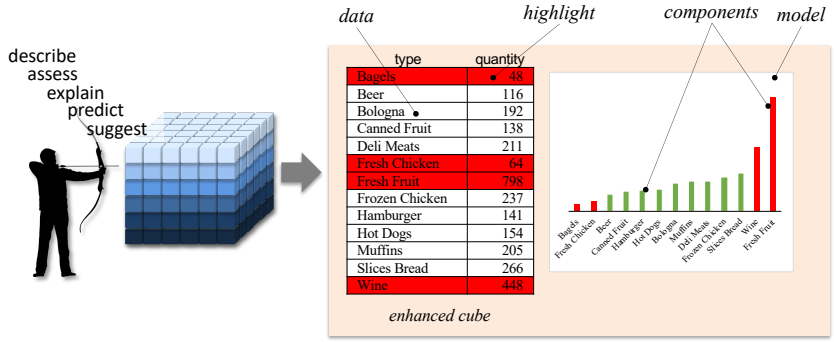



**Figure 1:** The IAM approach: the user expresses an intention and receives in return an enhanced cube

**Example 1.** *Let a* SALES *cube be given, and let the user's intention be:* with SALES describe quantity for month = '1997-04' by type using outliers. *Firstly, the subset of cells for April 1997 are selected from the* SALES *cube, aggregated by product type, and projected on measure* quantity *(in OLAP terms, a slice-and-dice and a roll-up operator are applied). Then, the outliers are found in these cells based on the values of* quantity. *Finally, a measure of interestingness is computed for the two components obtained (the outlier cells, and the non-outlier ones), and the cells belonging to the component with maximum interestingness (outlier cells) are highlighted in the results shown to the user (see Figure 1).* □

After introducing a formalism to manipulate cubes and queries in Section 2, in Section 3 we introduce models, components, and enhanced cubes. Then, in Section 4 we show how an intention is transformed into an execution plan, and in Section 5 we explain how enhanced cubes are visualized. Finally, in Section 6 we discuss the related literature and draw the conclusion.

## 2. Formalities

In this section we introduce the formal notations we will use in the paper to manipulate cubes. We start by defining cube schemata.

**Definition 1 (Hierarchy and Cube Schema).** *A* hierarchy *is a couple* $h = (L_h, \succeq_h)$ *where: (i)* $(L_h, \succeq_h)$ *is a* roll-up *total order of categorical* levels*; (ii) each level* $l \in L_h$ *is coupled with a* domain $Dom(l)$ *including a set of* members. *The top level of* $\succeq_h$ *is called* dimension. *A* cube schema *is a couple* $\mathcal{C} = (H, M)$ *where* $H$ *is a set of hierarchies and* $M$ *is a set of numerical measures, with each measure* $m \in M$ *coupled with one aggregation operator* $op(m) \in \{\mathtt{sum}, \mathtt{avg}, \ldots\}$.

**Example 2.** *For our working example it is* SALES $= (H, M)$*, where* $H = \{h_{\mathsf{Date}}, h_{\mathsf{Customer}}, h_{\mathsf{Product}}, h_{\mathsf{Store}}\}$, $M = \{\mathsf{quantity}, \mathsf{storeSales}, \mathsf{storeCost}\}$, date $\succeq$ month $\succeq$ year, customer $\succeq$ gender, product $\succeq$ type $\succeq$ category, store $\succeq$ city $\succeq$ country, $op(\mathsf{quantity}) = op(\mathsf{storeSales}) = op(\mathsf{storeCost}) = \mathtt{sum}$. □

Aggregation is the basic mechanism to query cubes, and it is captured by the following definition of group-by set.

**Definition 2 (Group-by Set and Coordinate).** *Given cube schema $\mathcal{C} = (H, M)$, a* group-by set *$G$ of $\mathcal{C}$ is a set of levels, at most one from each hierarchy of $H$. A* coordinate *of a group-by set $G$ is a tuple of members, one for each level of $G$.*

**Example 3.** *Two group-by sets of* SALES *are $G_1 = \{\text{date}, \text{type}, \text{country}\}$ and $G_2 = \{\text{month}, \text{category}\}$. Example of coordinates of these group-by sets are, respectively, $\gamma_1 = \langle$1997-04-15, Fresh Fruit, Italy$\rangle$ and $\gamma_2 = \langle$1997-04, Fruit$\rangle$.* □

The instances of a cube schema are called cubes and are defined as follows:

**Definition 3 (Cube).** *A* cube *over $\mathcal{C}$ is a tuple $C = (G_C, M_C, \omega_C)$ where: (i) $G_C$ is a group-by set of $\mathcal{C}$; (ii) $M_C \subseteq M$; (iii) $\omega_C$ is a partial function that maps some coordinates of $G_C$ to a numerical value for each measure $m \in M_C$.*

Each coordinate $\gamma$ that participates in $\omega_0$, with its associated tuple $t$ of measure values, is called a *cell* of $C$ and denoted $\langle \gamma, t \rangle$. A cube whose group-by set $G_C$ includes all and only the dimensions of the hierarchies in $H$ and such that $M_C = M$, is called a *base cube*, the others are called *derived cubes*. In OLAP terms, a derived cube is the result of either a roll-up, a slice-and-dice, or a projection made over a base cube; this is formalized as follows.

**Definition 4 (Cube Query).** *A* query *over cube schema $\mathcal{C}$ is a triple $q = (G_q, P_q, M_q)$ where: (i) $G_q$ is a group-by set of $H$; (ii) $P_q$ is a (possibly empty) set of selection predicates, each expressed over one level of $H$; (iii) $M_q \subseteq M$.*

**Example 4.** *The cube query over* SALES *used in Example 1 is $q = (G_q, P_q, M_q)$ where $G_q = \{\text{type}\}$, $P_q = \{\text{month} = \text{'1997-04'}\}$, and $M_q = \{\text{quantity}\}$. A cell of the resulting cube $q(\text{SALES}_0)$ (where $\text{SALES}_0$ is the base cube) is $\langle$*Canned Fruit$\rangle$ *with associated value 138 for* quantity. □

## 3. Enhancing cubes with models

Models are concise, information-rich knowledge artifacts [4] that represent relationships hiding in the cube cells. The possible models range from simple functions and measure correlations to more elaborate techniques such as decision trees, clusterings, etc. A model is bound to (i.e., is computed over the levels/measures of) one cube, and is made of a set of components (e.g., a clustering model is made of a set of clusters). In the IAM, a relevant role is taken by data-to-model mappings. Indeed, a model partitions the cube on which it is computed into two or more subsets of cells, one for each component (e.g., the subsets of cells belonging to each cluster).

**Definition 5 (Model and Component).** *A* model *is a tuple $\mathcal{M} = (t, alg, C, In, Out, \mu)$ where:*

*(i) $t$ is the model type;*

*(ii) $alg$ is the algorithm used to compute $Out$;*

*(iii)* $C$ *is the cube to which* $\mathcal{M}$ *is bound;*

*(iv)* $In$ *is the tuple of levels/measures of* $\mathcal{C}$ *and parameter values supplied to* $alg$ *to compute* $\mathcal{M}$*;*

*(v)* $Out$ *is the set of components that make up* $Out$*;*

*(vi)* $\mu$ *is a function mapping each coordinate of* $C$ *to one component of* $Out$*.*

*Each model* component *is a tuple of a component identifier plus a variable number of properties that describe that component.*

In the scope of this work, it is $t \in \{\text{top-k}, \text{bottom-k}, \text{skyline}, \text{outliers}, \text{clustering}\}$. For instance, for $t =$ clustering, each component is a cluster and is described by its centroid.

**Example 5.** *A possible model over the derived cube* $q(\mathsf{SALES}_0)$ *in Example 4 is characterized by* $t =$ *clustering,* $alg =$ *K-Means,* $C = q(\mathsf{SALES}_0)$*,* $In = \langle \mathsf{quantity}, n = 3, rndSeed = 0 \rangle$*,* $Out = \{c1, c2, c3\}$*,* $\mu(\langle \textit{Bagels} \rangle) = c1$*,* $\mu(\langle \textit{Beer} \rangle) = c1$*,* $\mu(\langle \textit{Bologna} \rangle) = c2$*, ..., where* $n$ *is the desired number of clusters and* $rndSeed$ *is the seed to be used by the k-means algorithm to randomly generate the 3 seed clusters. Component* $c1$ *is characterized by property* $centroid$ *with value 76.* □

As the last step in the IAM approach, cube $C$ is enhanced by associating it with a set of models bound to $C$ and with a *highlight*, i.e., with the subset of cells corresponding to the most interesting component of the model; these cells are determined via function $\mu$.

**Definition 6.** *An* enhanced cube $E$ *is a triple of a cube* $C$*, a set of models* $\{\mathcal{M}_1, \ldots, \mathcal{M}_r\}$ *bound to* $C$*, and a highlight* $c_{high} = argmax_{\{c \in \bigcup_{i=1}^{r} Out_i\}}(interest(c))$.

How to estimate the interestingness of component $c$, $interest(c)$, is explained in detail in [2]. Here we just mention that we consider three facets of interestingness identified in [5], namely, *novelty*, *peculiarity*, and *surprise*.

## 4. Execution plans for describe intentions

The describe operator provides an answer to the user asking "show me my business" by describing one or more cube measures, possibly focused on one or more level members, at some given granularity [1]. The cube is enhanced by showing either the top/bottom-k cells, the skyline, the outliers, or clusters of cells. Let $C_0$ be a base cube over cube schema $\mathcal{C} = (H, M)$; the syntax for describe is

$$\begin{aligned} &\text{with } C_0 \text{ describe } m_1, \ldots, m_z \text{ [ for } P \text{ ] [ by } l_1, \ldots, l_n \text{ ]} \\ &\quad \text{[ using } t_1 \text{ [ size } k_1 \text{ ]}, \ldots, t_r \text{ [ size } k_r \text{ ]]} \end{aligned}$$

(optional parts are in brackets) where $m_1, \ldots, m_z \in M$ are measures of $\mathcal{C}$, $P$ is a set of selection predicates each over one level of $H$, $\{l_1, \ldots, l_n\}$ denote a group-by set of $H$, $t_1, \ldots, t_r$ are

model types, and the $k_i$'s are the desired sizes to be applied to the models returned as explained in point 2 below.

The plan corresponding to a fully-specified intention, i.e., one where all optional clauses have been specified, is:

1. Execute query $q = (G_q, P_q, M_q)$, where $G_q = \{l_1, \ldots, l_n\}$, $P_q = P$, and $M_q = \{m_1, \ldots, m_z\}$. Let $C = q(C_0)$.
2. For $1 \leq i \leq r$, compute model $\mathcal{M}_i = (t_i, alg_i, C, In_i, Out_i, \mu_i)$ and for each $c \in Out_i$, compute $interest(c)$. Size $k_i$ is used for clustering to determine the number of clusters to be computed, for top-k and bottom-k to determine the number of cells to be returned, for outliers to determine the number of outliers; it is neglected for the skyline.
3. Find the highlight $c_{high} = argmax_{\{c \in \bigcup_i Out_i\}}(interest(c))$.
4. Return the enhanced cube $E$ consisting of $C$, $\{\mathcal{M}_1, \ldots \mathcal{M}_r\}$, and $c_{high}$.

Partially-specified intentions are interpreted as follows:

- If the for clause has not been specified, we consider $P_q = TRUE$.
- If the by clause has not been specified, we consider $G_q = \emptyset$.
- If the using $t_1, \ldots, t_r$ clause has not been specified, all model types listed in Section 3 are computed over $C$ (the skyline is computed only if $z > 1$, i.e., at least two measures have been specified).
- If the size clause has not been specified for one or more models, the value of $k_i$ is determined automatically through the Elbow method.

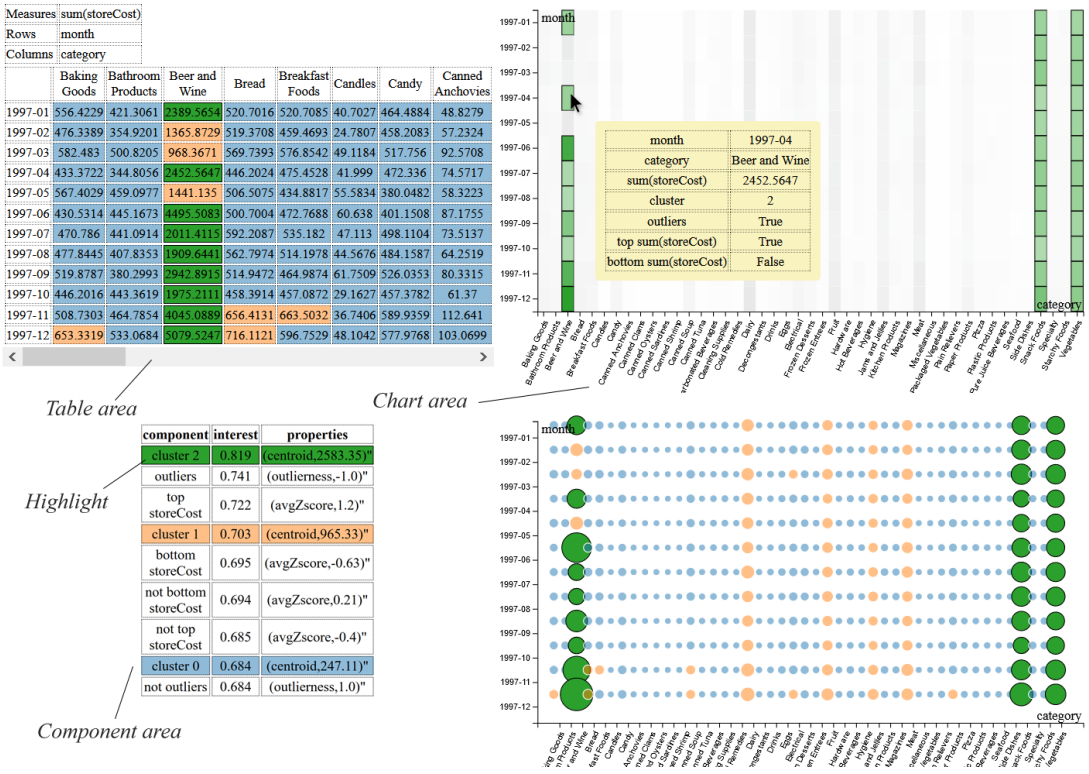
**Example 6.** *Consider the following session on the* SALES *cube:*

with SALES describe quantity for month = '1997-04' by type
with SALES describe quantity by category using clustering size 3

*The models computed for the first intention are top-k, bottom-k, clustering, and outliers (computing the skyline for a single measure makes no sense). For the second intention, a clustering producing 3 clusters is computed.* □

## 5. Visualizing enhanced cubes

To provide an effective description of an enhanced cube we couple text-based and graphical representations with an ad-hoc interaction paradigm. Specifically, the visualization includes three distinct but inter-related areas: a *table* area that shows the cube cells using a pivot table; a *chart* area that complements the table area by representing the cube cells through one or more charts; a *component* area that shows a list of model components sorted by their interestingness. The guidelines adopted to select the charts are detailed in [2]. The interaction paradigm we adopt is component-driven. Specifically, clicking on one component $c$ in the component area leads to emphasize the corresponding cube cells (i.e., those that map to $c$ via function $\mu$) both




| | Baking Goods | Bathroom Products | Beer and Wine | Bread | Breakfast Foods | Candles | Candy | Canned Anchovies |
|---|---|---|---|---|---|---|---|---|
| 1997-01 | 556.4229 | 421.3061 | 2389.5654 | 520.7016 | 520.7085 | 40.7027 | 464.4884 | 48.8279 |
| 1997-02 | 476.3389 | 354.9201 | 1365.8729 | 519.3708 | 459.4693 | 24.7807 | 458.2083 | 57.2324 |
| 1997-03 | 582.483 | 500.8205 | 968.3671 | 569.7393 | 576.8542 | 49.1184 | 517.756 | 92.5708 |
| 1997-04 | 433.3722 | 344.8056 | 2452.5647 | 446.2024 | 475.4528 | 41.999 | 472.336 | 74.5717 |
| 1997-05 | 567.4029 | 459.0977 | 1441.135 | 506.5075 | 434.8817 | 55.5834 | 380.0482 | 58.3223 |
| 1997-06 | 430.5314 | 445.1673 | 4495.5083 | 500.7004 | 472.7688 | 60.638 | 401.1508 | 87.1755 |
| 1997-07 | 470.786 | 441.0914 | 2011.4115 | 592.2087 | 535.182 | 47.113 | 498.1104 | 73.5137 |
| 1997-08 | 477.8445 | 407.8353 | 1909.6441 | 562.7974 | 514.1978 | 44.5676 | 484.1587 | 64.2519 |
| 1997-09 | 519.8787 | 380.2993 | 2942.8915 | 514.9472 | 464.9874 | 61.7509 | 526.0353 | 80.3315 |
| 1997-10 | 446.2016 | 443.3619 | 1975.2111 | 458.3914 | 457.0872 | 29.1627 | 457.3782 | 61.37 |
| 1997-11 | 508.7303 | 464.7854 | 4045.0889 | 656.4131 | 663.5032 | 36.7406 | 589.9359 | 112.641 |
| 1997-12 | 653.3319 | 533.0684 | 5079.5247 | 716.1121 | 596.7529 | 48.1042 | 577.9768 | 103.0699 |

**Figure 2:** The visualization obtained for the intention in Example 7

in the table area and in the chart area. The highlight is the top component in the list and is selected by default. Following the *details-on-demand* paradigm [6], interaction is enhanced using a tooltip that, when the mouse is positioned on a data point, shows its coordinate, its measure value(s), and the component(s) it belongs to.

**Example 7.** *Figure 2 shows the visualization obtained when the following intention is formulated:* with SALES describe storeCost by month, category. *On the top-left, the table area; on the right, the chart area; on the bottom-left, the component area. Here a heat map and a bubble chart have been selected. The top-interestingness component is a cluster, so a color has been assigned to each component of clustering (i.e., to each cluster) and is uniformly used in all three areas. The highlight (in green) is currently selected and is emphasized using a thicker border in all areas. A tooltip with all the details about a single cell is also shown (in yellow).* □

## 6. Related work and conclusion

The idea of coupling data and analytical models was born in the 90's with inductive databases, where data were coupled with patterns meant as generalizations of the data. Later on, data-to-model unification was addressed in MauveDB [7], which provides a language for specifying model-based views of data using common statistical models.

The coupling of the OLAP paradigm and data mining to create an approach where concise patterns are extracted from multidimensional data for user's evaluation, was the goal of some approaches commonly labeled as OLAM [8]. In this context, k-means clustering is used by [9] to dynamically create semantically-rich aggregates of facts other than those statically provided by dimension hierarchies. Similarly, the shrink operator is proposed by [10] to compute small-size approximations of a cube via agglomerative clustering. Other operators that enrich data with knowledge extraction results are DIFF [11], which returns a set of tuples that most successfully describe the difference of values between two cells of a cube, and RELAX [12], which verifies whether a pattern observed at a certain level of detail is also present at a coarser level of detail, too. Finally, [13] reuse the OLAP paradigm to explore prediction cubes, i.e., cubes where each cell summarizes a predictive model trained on the data corresponding to that cell. The IAM approach can be regarded as OLAM since, like the approaches mentioned above, it relies on mining techniques to enhance the cube resulting from an OLAP query. However, while each of the approaches above uses one single technique (e.g., clustering) to this end, the IAM leans on multiple mining techniques to give users a wider variety of insights, using the interestingness measure to select the most relevant ones.

To the best of our knowledge, though some tools (e.g., Spotfire and Tableau) integrate OLAP and analytics capabilities in the same environment, none of them allows users to formulate queries at a higher level of abstraction than OLAP (as done in the IAM using intentions), nor they support the automated *out-of-the-box* enrichment of cubes with insights obtained by analytics (as done in the IAM through enhanced cubes).

In this paper we have given a proof-of-concept for the IAM vision by delivering an implementation of the describe operator, relying on a visual metaphor to display enhanced cubes. Our implementation uses a simple multidimensional engine [14, 15] that relies on the Oracle 11g DBMS to execute queries on a star schema; the mining models are imported from the Scikit-Learn Python library. The web-based visualization is implemented in JavaScript and uses the D3 library. The prototype can be accessed at http://semantic.csr.unibo.it/describe/.

In [2], we have showed that our approach diminishes the effort for formulating complex analyses while ensuring that performances are compatible with near-real-time requirements of interactive sessions. Specifically, using the ASCII character length as an approximation for the effort it takes to craft a query, we evaluated the saving in user's effort when writing a describe intention over the one necessary to obtain the same result using plain SQL and Python. We considered a simple session including three intentions, where the by clause is progressively enlarged and all the models are computed. Remarkably, it turned out that the total formulation effort using SQL+Python is about two orders of magnitude larger than using describe intentions (in the average, about 5400 vs. 55 chars). For the efficiency test we used the FoodMart data (github.com/julianhyde/foodmart-data-mysql) and the same session mentioned above. Table 1 shows the total execution time and its breakdown into the times necessary to query the base cube, to compute the models, to measure the interestingness, and to generate the pivot table returned to the browser. Remarkably, it turns out that at most 18 seconds are necessary to retrieve and visualize an enhanced cube of more than 86000 cells, which is perfectly compatible with the execution time of a standard OLAP query.

The main directions for future research we wish to pursue are: (i) evaluate the usability of the approach by conducting tests with real users, and (ii) extend the approach to operate with

**Table 1**
Execution times in seconds for three intentions with increasing cardinalities of $C$ (the tests were run on an Intel Core(TM)i7-6700 CPU@3.40GHz with 8GB RAM)

| *Intention* | $\|C\|$ | *Query* | *Model* | *Interestingness* | *Pivot* | *Total* |
|---|---|---|---|---|---|---|
| $I_1$ | 323 | 0.10 | 0.25 | 0.00 | 0.00 | 0.36 |
| $I_2$ | 20525 | 0.22 | 5.90 | 0.36 | 0.36 | 6.83 |
| $I_3$ | 86832 | 0.22 | 8.50 | 7.43 | 1.72 | 17.87 |

*dashboards* of enhanced cubes.

## References


[1] P. Vassiliadis, P. Marcel, S. Rizzi, Beyond roll-up's and drill-down's: An intentional analytics model to reinvent OLAP, Inf. Sys. 85 (2019) 68–91.

[2] M. Francia, P. Marcel, V. Peralta, S. Rizzi, Enhancing cubes with models to describe multidimensional data, Inf. Sys. Frontiers 24 (2022) 31–48.

[3] M. Francia, M. Golfarelli, P. Marcel, S. Rizzi, P. Vassiliadis, Assess queries for interactive analysis of data cubes, in: Proc. of EDBT, 2021, pp. 121–132.

[4] M. Terrovitis, P. Vassiliadis, S. Skiadopoulos, E. Bertino, B. Catania, A. Maddalena, S. Rizzi, Modeling and language support for the management of pattern-bases, Data Knowl. Eng. 62 (2007) 368–397.

[5] P. Marcel, V. Peralta, P. Vassiliadis, A framework for learning cell interestingness from cube explorations, in: Proc. of ADBIS, 2019.

[6] B. Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations, in: Proc. of IEEE Symp. on Visual Languages, 1996, pp. 336–343.

[7] A. Deshpande, S. Madden, MauveDB: supporting model-based user views in database systems, in: Proc. of SIGMOD, 2006, pp. 73–84.

[8] J. Han, OLAP mining: Integration of OLAP with data mining, in: Proc. of Working Conf. on Database Semantics, 1997, pp. 3–20.

[9] F. Bentayeb, C. Favre, RoK: Roll-up with the k-means clustering method for recommending OLAP queries, in: Proc. of DEXA, 2009, pp. 501–515.

[10] M. Golfarelli, S. Graziani, S. Rizzi, Shrink: An OLAP operation for balancing precision and size of pivot tables, Data Knowl. Eng. 93 (2014) 19–41.

[11] S. Sarawagi, Explaining differences in multidimensional aggregates, in: Proc. of VLDB, 1999, pp. 42–53.

[12] G. Sathe, S. Sarawagi, Intelligent rollups in multidimensional OLAP data, in: Proc. of VLDB, 2001, pp. 531–540.

[13] B. Chen, L. Chen, Y. Lin, R. Ramakrishnan, Prediction cubes, in: Proc. of VLDB, 2005, pp. 982–993.

[14] M. Francia, E. Gallinucci, M. Golfarelli, Towards conversational OLAP, in: Proc. of DOLAP, 2020, pp. 6–15.

[15] M. Francia, E. Gallinucci, M. Golfarelli, COOL: A framework for conversational OLAP, Inf. Syst. 104 (2022) 101752.