Research Directions of OLAP Personalizaton

Natalija Kozmina, Laila Niedrite

Faculty of Computing, University of Latvia, Riga LV-1586, Latvia

{natalija.kozmina, laila.niedrite}@lu.lv

Abstract. In this paper we have highlighted five existing approaches for introducing personalization in OLAP: preference constructors, dynamic personalization, visual OLAP, recommendations with user session analysis and recommendations with user profile analysis and have analyzed research papers within these directions. We have provided an evaluation in order to point out i) personalization options, described in these approaches, and its applicability to OLAP schema elements, aggregate functions, OLAP operations, ii) the type of constraints (hard, soft or other), used in each approach, iii) the methods for obtaining user preferences and collecting user information. The goal of our paper is to systematize the ideas proposed already in the field of OLAP personalization to find out further possibility for extending or developing new features of OLAP personalization.

Keywords: OLAP personalization, user preferences.

1 Introduction and Related Work

The OLAP applications are built to perform analytical tasks within large amount of multidimensional data. During working sessions with OLAP applications the working patterns can be various. Due to the large volumes of data the typical OLAP queries performed via OLAP operations by users may return too much information that sometimes makes further data exploration burdening or even impossible. In case of too many constraints chosen the result set can be empty. In other cases, when the user tries to explore previously unknown data, the OLAP query result may highly differ from expectations of the user. Thus, the user is rather limited in expressing his/her intentions or likes and dislikes in order to get more satisfying results.

A query personalization method that takes user likes and dislikes into consideration exists in traditional databases [1]. So, in case of executing a personalized query, the user gets more appropriate results. Similar ideas seem attractive also for research in the data warehousing field and the topicality of this issue is demonstrated in the recent works of many authors on data warehouse personalization.

There are various aspects of data warehouse personalization.

Data warehouse can be personalized at the schema level. [2] use the data warehouse multidimensional model, user model and rules for the data warehouse personalization. As a result, a data warehouse user is able to work with a personalized OLAP schema, which matches his needs best of all.

Users may express their preferences on OLAP queries [3]. In this case, the problem of performing time-consuming OLAP operations to find the necessary data can be significantly improved.

One of the methods of personalizing OLAP systems is to provide query recommendations to data warehouse users. OLAP recommendation techniques are proposed in [4] and [5]. In [4] former sessions of the same data warehouse user are being investigated. User profiles that contain user preferences are taken into consideration in [5], while generating query recommendations.

Other aspect of OLAP personalization is visual representation of data. [6, 7] introduce multiple layouts and visualization techniques that might be interactively used for different analysis tasks.

Our experience in using standard applications for producing and managing data warehouse reports in the University of Latvia (UL) as well as participation in scientific projects and development of our own data warehouse report management tool [8] served as a motivation for further studies in the field of OLAP personalization. It has been stated that both tools (standard and newly-developed) allow defining ad-hoc queries, displaying reports as tables and graphs and analyzing data using hierarchies. Users with administrator rights may modify other user right for data warehouse report creating, exploring and editing. A user may adjust visual representation of the workbook, which contains generated reports (e.g. change font color and style, etc.). Since options to personalize data warehouse reports by means of these tools are currently very limited, we consider the report management tool, developed in the UL, to be an experimental environment for introducing OLAP personalization.

As stated in [3], OLAP preferences deserve more attention by researchers. In this paper an overview of different OLAP personalization approaches is presented. The goal of our paper is to classify the ideas that have been already proposed in this field in order to find questions that still remain unanswered.

The rest of the paper is organized as follows: section 2 introduces a review of existing OLAP personalization types and its evaluation; section 3 discusses hard and soft constraints in user preferences as well as methods for gathering user information and obtaining user preferences; section 4 concludes the paper.

2 OLAP Personalization Types

To the best of our knowledge, there are various OLAP personalization types – OLAP schema personalization, personalization during runtime, visual personalization of query results, etc. – which are briefly described in this section. A comparison, which includes personalization types and OLAP schema elements and operations, will follow. Proposed comparison gives our evaluation of personalization described by indicating, whether personalization of certain type is applicable to OLAP schema elements and operations, or not.

2.1 Description of OLAP Personalization Approaches

The first approach to be considered is OLAP schema personalization with *Preference Constructors (PC)*. An algebra that allows formulating of preferences on attributes, measures and hierarchies is defined in [3]. An important feature of proposed algebra is an opportunity to express preferences for hierarchy attributes of group-by sets, which consequently leads to expressing preferences for facts. Rollup function is used to outspread preferences applied to attributes along the whole hierarchy. Preferences can be defined on both attributes and measures, i.e. on categorical or numerical attributes.

Consider two kinds of preferences: base and complex [3]. Base preference constructors are applied to attribute, measure, hierarchy level. Complex preferences consist of combination of base preferences, which can be expressed by means of formal grammar. Base preference constructor in this grammar is one of predefined operators like POS, NEG, BETWEEN or some others.

The next approach is *Dynamic Personalization (DP)*. The time and method of creation of an adapted OLAP cube define the type of personalization – static or dynamic. Static OLAP personalization means that for different users of the data warehouse diverse OLAP cubes are created during design time. Dynamic OLAP personalization means that an adapted OLAP cube is created during the execution time according to the needs and performed actions of the user. [2] cover dynamic OLAP personalization, because it is a more complicated task as it involves explicit or implicit interaction with user. Based on ECA-rules (see [9]), PRML (described in [10]) is used in [2] for specification of OLAP personalization rules. The structure of such PRML rule can be presented with following statement:

`when` *`event`* `do if` *`condition`* `then` *`action`* `endIf endWhen.`

There are two kinds of actions proposed to be used in personalization rules in [2]. In order to get information about the user during runtime and update the user model or to update values of dimension attributes and cube measures, a *set*-action is used (e.g. for calculating user's degree of interest in certain dimension attributes). To personalize multidimensional model, *hide*-actions are used on OLAP schema objects (e.g. a *hide*-action may be executed, if the user's degree of interest in a certain dimension attribute is lower than a pre-defined value).

Visual personalization of OLAP cube – *Visual OLAP (VO)* – may also be considered as a personalization action. The concept of Visual OLAP is disburdening the user from composing queries in "raw" database syntax (SQL, MDX), whereas events like clicking and dragging are transformed into valid queries and executed [7]. In [5, 6, 11] authors present a user interface for OLAP, where user is explicitly involved. In [6] users are able to navigate in dimensional hierarchies using a schema-based data browser, whereas in [5, 11] users are provided with an interface for formulating queries by means of manipulation with graphical OLAP schema and rules. The query is composed by the user when he/she selects a measure and an aggregation function [6]. Dimensions for “drilling down” are chosen and the values are set as filters. Having selected the measure and the aggregate function, the user simply drags any dimension folder into the visualization area to create a new level in the *decomposition tree*. The decomposition tree is gained from an aggregate measure as a root, splitting it along chosen dimensions. Different layouts for decomposition trees are proposed in [6].

The last two approaches for personalization in OLAP to be considered are based on providing query recommendations to the user by means of *User Session Analysis (RUSA)* and *User Preference Analysis (RUPA).*

The idea of *RUSA* is described in [4], where users’ previous data analysis patterns using OLAP server query log during sessions are taken into consideration. Cube measure values are being compared and a significant unexpected difference in the data is being detected. The emphasis is not on recommending queries from sessions that are prior to the current session, but on recommending queries from all sessions, where user found the same unexpected data as in current session. In this approach user preferences are not taken into consideration. A concept of difference query for rollup and drill-down operations as a query whose result confirms the difference of measure values at a higher level of detail for rollup or lower level of detail is introduced by [4]. Authors analyze user queries, executed during users’ sessions, thus we consider that personalization is applicable to OLAP select operation.

RUPA approach is presented in [5], where a context-based method for providing users with recommendations for further exploration is proposed. An analysis context includes two disjoint set elements (i.e. a set of OLAP schema elements – cubes, measures, dimensions, attributes, etc. and a set of its values), which are represented in a tree structure (though visualized as a multidimensional table). Also, restriction predicates i.e. restrictions on measures (associated with an aggregate function) or conditions on dimension attributes are included into analysis context. Both types of user preferences – schema- and content-level preferences – are stated in the user profile and ranked with relevance score (a real number in the range [0; 1]). The idea of ranking preferences is also mentioned in [11]. User preferences later on are used in generating recommendations, filtering a recommendation with the highest overall score and displaying it to the user. Preferences in user profiles are also used for comparing queries and personalizing query result visualization in [12].

2.2 Comparison of Existing Approaches for OLAP Personalization

We analyzed and compared all previously described approaches to give an overview on applying personalization of different type to OLAP schema elements, functions and typical OLAP operations. The results are given in Table 1. One axis of the table contains the main concepts of OLAP systems: OLAP schema elements, aggregate functions, OLAP operations. The OLAP schema elements – dimensions and its attributes, hierarchies and its levels, cubes (or fact tables) and its measures – are described a lot in the literature, also in [13]. Aggregate functions are described in

[14]. OLAP operations slice and dice (or select), drilldown, rollup and pivot (or rotate) are described in [13, 15, 16]. In our comparison we use a term *select* instead of *slice and dice* for the sake of simplicity, because some of the personalization types provide personalization of SQL-like select-queries. Also, here we use the term *rotate* instead of *pivot*. The second axis of the table contains all previously described personalization types. The cells of the table contain a value from a set of acronyms to represent our evaluation: “A” – applicable: personalization applicability to OLAP schema element, aggregate function or OLAP operation is explicitly defined by the authors of articles on *PC*, *DP*, *VO*, *RUSA* and *RUPA*; “D” – derivable: personalization applicability to OLAP schema element, aggregate function or OLAP operation can be derived, taking into account other personalization aspects, which are presented in the paper (e.g. personalization considers rollup operation, but drilldown operation is not mentioned in the paper; in that case we say that personalization considering drilldown is *derivable*, because drilldown operation is an inverse operation or rollup, etc.); “-” – there is no information; personalization applicability to OLAP schema element, aggregate function or OLAP operation is not described in the paper.

Table 1. Applicability of different personalization types to OLAP objects.

Pers. Type / Pers. Object	**Dimension**	**Dimension attribute**	**Hierarchy**	**Hierarchy level**	**Cube**	**Cube measure**	**Aggregation function**	**Select**	**Drilldown**	**Rollup**	**Rotate**
PC	-	*A*	-	*A*	-	*A*	-	*D*	*D*	*A*	-
DP	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A**, *D***	*A*	*A*	*A**, -**
VO	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*
RUSA	*A*	*A*	*A*	*A*	*A*	*A*	*D*	*A*	*A*	*A*	-
RUPA	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*D*	*D*	-

In *DP* the extent of personalization applied to certain OLAP operations varies, depending on the approach, proposed by different authors. Authors [11] are marked with “*”, authors [2] – with “**”.

One may observe that personalization of OLAP schema elements is mostly present in all proposed OLAP personalization types, except for preference constructors (*PC*), where the way of expressing user preferences for dimensions, hierarchies, cubes as whole as well as aggregate functions, is not described. However, preferences on OLAP operations such as Select, Drilldown and Rollup are not always expressed explicitly and there is a lack of information about personalization options, considering Rotate OLAP operation.

3 A Closer Look at User Preferences: Hard and Soft Constraints

Although the role of the preferences was recognized in applications long ago, the database researchers paid attention to this issue only around year 2000 [17, 18, 19, 20]. It was observed that in database queries WHERE-conditions are *hard constraints* and either the non-empty result set is returned if all the conditions are satisfied, or an empty set is returned in the opposite case. Queries with hard constraints either deliver exactly the desired object if it exists, or reject the user's request otherwise [21].

The authors of [22] define *soft constraints* as functions that map any potential value assignment into a numerical value that indicates the preference that this value or value combination carries. In information retrieval soft constraints are used and results are arranged, according to its relevancy to initial query conditions. The difference between hard and soft constraints is that soft constraints can be evaluated, whereas hard constraints can be either satisfied or not. User preferences express soft constraints. Eventually, different approaches to use soft constraints in database queries have appeared [19, 20], turning database queries into "preference queries". In papers [19, 21, 23, 24] an implementation of the framework using Preference SQL is described, which is translated to SQL, and used in several deployed applications. [19, 21] point out that extending SQL by preferences will enable better personalized search to gain more targeted results.

Preference SQL consists of Standard SQL constructs and preferences [24]. Preference queries are specified in [23] using a SELECT-FROM-WHERE part (standard SQL; the WHERE-clause specifies hard constraints) and a PREFERRING-GROUPING part (expresses preferences i.e. soft constraints) of a query. In both parts of a preference query AND can be used to combine more than one constraint, but in the PREFERRING clause it has a meaning of Pareto operator. In this case AND prescribes combination of equally important preferences.

Our purpose is to understand, what kind of user preferences can be expressed in each of OLAP personalization types earlier discussed. We consider the hard and soft constraints as a means to express the user preferences. In the OLAP domain the definition of *preference* is proposed by [3], stating that a preference is a couple of two operators, where first states that one fact value in OLAP schema is preferred to the other, but the other operator states that facts are equivalent (or *substitutable* [25]).

Table 2 illustrates, which method is applied; "+" ("-") indicates that a method is (isn't) applied in each of personalization types: Hard/Soft Constraints or Other (meaning that the method used cannot be categorized as hard/soft constraints).

Table 2. OLAP Personalization types and applied constraints.

Personalization Type / Method	**Hard Constraints**	**Soft Constraints**	**Other**
PC	-	+	-
DP	+	-	-
VO	+	-	-
RUSA	-	-	+
RUPA	-	+	-

Preference Constructors (*PC*) use soft constraints as there is a possibility to express user's likes and dislikes, e.g. a user would like to obtain student activity data (i.e. time spent on exploring course informational resources, quantity of tasks assigned and completed, grades for completed tasks, etc.) considering course, named "*Data Warehouses*", which is an attribute of Course dimension in data warehouse.

Example 1. Consider a hierarchy Course $\succ_H$ Study Program $\succ_H$ Faculty, where $\succ_H$ is a Rollup function over hierarchy H. *Biology Masters* is one of study programs, belonging to the *Faculty of Biology*. NEG(StudyProgram, "Biology Masters") states that data that does not map to *Biology Masters* study program, does not refer to courses of *Biology Masters* study program and does not map to the *Faculty of Biology*, is preferred to all the other data.

One of the aspects of Visual OLAP (*VO*) is user browsing through navigational OLAP schema and filtering the OLAP schema objects to be displayed [7]. Users' navigation events such as clicking and dragging are translated to valid SQL-queries with WHERE-clause, which in fact is a hard constraint in standard SQL [21].

We consider that there are hard constraints in dynamic personalization (*DP*) with ECA-rules as the sets of operations with both numerical and non-numerical attributes in condition-part of ECA-rules are the same as operations, included in hard constraints. In the following example "=" operation is used when checking, whether the data warehouse user role is "Student" or not; if the user is a student, then attribute *BusinessTrip* of the dimension *Person* is being hidden.

Example 2. **Rule**: hideBusinessTrip
When SessionStart **Do If** (User.Role = "Student")
Then hideDescriptor(Person.BusinessTrip) **EndIf EndWhen**

The main idea of query recommendations approach, based on investigation of user sessions (*RUSA*), is to find unexpected difference in the data and generate further recommendations with the same unexpected data as the current session.

Example 3. If there is a difference that is a drop of the sales of some kind of product from 2009 to 2010, then recommended queries will contain the same difference in values. We consider that neither soft nor hard constraints are used in this type of personalization. Authors [4] use the technique that develops the ideas of DIFF operator, proposed in [26] and used for explaining reasons for sudden drops or increases in data values.

We consider that in user profiles, utilized for generation of recommendations (*RUPA*), soft constraints appear. A user may express the extent of liking or disliking as there is a relevance score that is associated with analysis element of OLAP schema [5]. Following example illustrates the usage of soft constraints in *RUPA*:

Example 4. P^{Role} = ('Role ≠ Guest'; 0.9; c), where 'Role ≠ Guest' is a predicate, which is a condition on dimension data (in other case a predicate may be a restriction on fact table data), 0.9 is a real number (between 0 and 1) that indicates relevance degree (a number closer to 0 means 'less relevant' and closer to 1 means 'more relevant'), c is an analysis context, which includes analyzed cube measures (with aggregate functions applied) and analysis axis (dimension/attribute). Here c = "Activity, Time/Date ≥ '01/01/2010'", which means that measures of Activity cube are analyzed and Time/Date is an analysis axis. P^{Role} = ('Role ≠ Guest'; 0.9; c) means that user's interest to include condition 'Role ≠ Guest' into qualification of user activity in course management system is very high.

3.1 Collecting User Data: Explicit and Implicit Approaches

Typically there are two approaches of collecting information about the user – explicit and implicit feedback [27]. Also, hybrid (i.e. explicit and implicit method combined) is possible.

Methodologies for *explicit* user information gathering are based on information input by users about themselves and their interests. Users enter information manually or choose pre-defined values from list. The problems arise, because the users are not always ready to give such information. In this case an explicit user profile could not be built. Also, [27] points out that user may not be very accurate, when providing information. User preferences may change over time, thus information in the profile may become out of date.

User profiles may be built based on *implicitly* gathered information. Implicit feedback gives us behavioral information about the user. Implicit feedback can be found by analyzing server logs, search and browsing history. A research on acquiring user preferences, based on implicit feedback, is presented in [28].

The most attractive aspect of the implicit feedback is that data about the user can be gathered without the presence of the user [27]. However, authors [27] point out some limitations of the implicit feedback. The data, observed by the user, is not always connected with an intention to observe it. Often the time when the data is displayed to the user is interpreted as reading time. Also, the user is unable to give negative feedback, to express negative interest or dislike, whereas mouse clicks are treated as positive feedback [27]. Sometimes during the search for essential information user clicks on unnecessary links, therefore, in many cases user activity could not be equalized to the count of clicks.

3.2 Methods for Obtaining User Preferences

[29] gives an overview of existing methods for extracting user preferences and giving further recommendations; [30] supplements the list with two more methods (*questions & answers*, *mixed initiative*):

- *Questions & Answers (Q&A)*. Information for user profile is collected, when user answers to the questions or fills in the form. The information in user profile stays unchanged, until the user updates it.
- *Mixed initiative (MI)*. This method is also called *candidate/critique mode*. Preferences are gained by proposing existing solutions to a user and receiving user evaluation. The solution is improved, according to the critique and proposed to the user again until it satisfies the user. An example of a system with implemented mixed initiative approach is a system, presented by [31], where an agent is implemented for the gathering user preferences when the user expresses his attitude to the observed data.
- *Content-based (CB)*. This method is used to generate recommendations from user preferences on other objects' features that user has already rated. Content-

based user profiles are updated, when new user preference-related information appears.

- *Utility and Knowledge-based (UKB)*. These methods make recommendations, based on similarity between what user needs and what is available.
- *Collaborative (C)*. In terms of this method multiple user ratings are aggregated and compared with the rating of a particular user of a certain object. As a result new recommendations are proposed to the user.
- *Demographic (D)*. This method is used to provide recommendations based on demographic characteristics of the user. Users with similar characteristics are grouped into classes.

Table 3 illustrates, which preference obtaining method is applied in each of five considered OLAP personalization approaches as well as demonstrates, how user information was collected – explicitly or implicitly.

Although in [3] OLAP preference algebra is proposed and technical implementation of preference constructors (*PC*) and its application is not described, we consider that the user would express the preferences explicitly. For instance, user may choose some out of the set of possible preference constructors and OLAP schema elements that serve as parameters for preference constructors, and assign values for OLAP schema elements (entering manually or choosing from range). Such approach is similar to *Q&A* method. Also, *UKB* method is being partly used, when, for instance, user states a certain attribute value in POS or NEG constructor, and then preferences are propagated over all levels of the corresponding hierarchy (see example 1).

Table 3. Preference obtaining and user information collection methods, used in different types of OLAP personalization.

	Preference Obtaining Method						***User Information Collection Method***	
Personalization Type	**Q&A**	**MI**	**CB**	**UKB**	**C**	**D**	**Explicit**	**Implicit**
PC	+	-	-	+	-	-	+	-
DP	-	-	+	+	-	-	-	+
VO	-	-	+	-	-	-	+	-
RUSA	-	-	-	+	-	-	-	+
RUPA	-	-	+	-	-	-	+	-

We suppose that there is a content-based (*CB*) approach used in dynamic personalization (*DP*). For instance, when ECA-rules are being executed, user context is taken into consideration e.g. user role in data warehouse (see example 2). Also, *UKB* approach is used, when user behavior is being analyzed, for instance, a utility is used for calculating user interest degree in certain aggregated data. Dynamic personalization uses an implicit method for collecting user information.

A content-based approach is also used in visual OLAP (*VO*) and in recommendations with user profile analysis (*RUPA*). In *VO* the user is able to move through navigational schema and set preferences for OLAP schema objects to be displayed (for example, choosing dimensions, setting constraints on dimension attribute values, etc.). In *RUPA* schema- and content-level preferences are stated in a user profile and ranked with relevance score. In both cases information is provided explicitly by the user.

In recommendations with user session analysis (*RUSA*) user information is gathered implicitly. To define user preferences, *UKB* approach is used – user previous session queries are being examined and a utility function, conceptually similar to DIFF operator, is applied (see example 3).

4 Conclusions and Future Work

In this paper we have highlighted five approaches for introducing personalization in OLAP: preference constructors (*PC*), dynamic personalization (*DP*), visual OLAP (*VO*), recommendations with user session analysis (*RUSA*) and recommendations with user profile analysis (*RUPA*). We do not claim that this is an exhaustive set of approaches for OLAP personalization and assume that it may be widened.

We have provided an evaluation in order to point out i) personalization options, described in these approaches, and its applicability to OLAP schema elements, aggregate functions, OLAP operations, ii) the type of constraints (hard, soft or other), used in each approach, iii) the methods for obtaining user preferences and collecting user information.

Comparing options of personalization application to personalization types, we may conclude that personalization of OLAP schema elements is mostly present in all proposed OLAP personalization types, except for preference constructors (*PC*), where the way of expressing user preferences for dimensions, hierarchies, cubes as whole as well as aggregate functions, is not described. Speaking about OLAP operations, we may notice that three out of four OLAP operations in three out of five personalization types are described implicitly (i.e. Select and Drilldown operations in *PC*, Select operation in *DP*, Drilldown and Rollup operations in *RUPA*). The information about expressing user preferences on Rotate operation is missing in all approaches, except for *DP* and *VO*. Thus, more attention should be drawn to user preferences for OLAP operations.

We proposed to group personalization types, according to the kind of constraint (soft, hard or other) that is used for expressing and managing user preferences. As a result, hard constraints are used in *DP* and *VO*, soft constraints – in *PC* and *RUPA* and other type of constraint (difference function) – in *RUSA*.

We analyzed applicability of existing methods for extracting user preferences [29, 30] and highlighted, how the user information is being collected (explicitly or implicitly). We may conclude that three out of six preference obtaining methods (i.e. questions & answers, content-based and utility & knowledge-based) are applied in considered types of personalization and the remaining three methods (mixed initiative, collaborative and demographic) are not applied. However, we assume that it is worthwhile to involve collaborative method for generating recommendations of queries, based on similarity of users' likes and dislikes.

We have taken the ideas of *RUPA* approach as a basis for our future work. We also proposed to involve collaborative method for generating recommendations of queries, based on similarity of users' likes and dislikes. A new method, which provides exhaustive description of interaction between user and data warehouse, is a subject of a separate paper that is currently being reviewed.

Acknowledgments This work has been supported by ESF project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

1. Koutrika, G., Ioannidis, Y. E.: Personalization of queries in database systems. In Proceedings of 20th International Conference on Data Engineering (ICDE'04), Boston, MA, USA, March 30 – April 2, 2004, pp. 597-608.
2. Garrigós, I., Pardillo, J., Mazón, J.-N., Trujillo, J.: A Conceptual Modeling Approach for OLAP Personalization. Conceptual Modeling - ER 2009, LNCS, vol. 5829, Springer, Heidelberg, 2009, pp. 401-414.
3. M. Golfarelli, S. Rizzi: Expressing OLAP Preferences. Berlin / Heidelberg, LNCS, vol. 5566/2009, Scientific and Statistical Database Management, 2009, pp. 83-91.
4. Giacometti, A., Marcel, P., Negre, E., Soulet, A.: Query Recommendations for OLAP Discovery Driven Analysis. In Proceedings of 12th ACM International Workshop on Data Warehousing and OLAP (DOLAP'09), Hong Kong, November 6, 2009, pp. 81-88.
5. Jerbi, H., Ravat, F., Teste, O., Zurfluh, G.: Preference-Based Recommendations for OLAP Analysis. In Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'09), Linz, Austria, August 31 – September 4, 2009, pp. 467-478.
6. Mansmann, S., Scholl, M. H.: Exploring OLAP Aggregates with Hierarchical Visualization Techniques. In Proceedings of 22nd Annual ACM Symposium on Applied Computing (SAC'07), Multimedia & Visualization Track, March 2007, Seoul, Korea, pp. 1067-1073.
7. Mansmann, S., Scholl, M. H.: Visual OLAP: A New Paradigm for Exploring Multidimensonal Aggregates. In Proceedings of IADIS International Conference on Computer Graphics and Visualization (MCCSIS'08), Amsterdam, The Netherlands, 24 - 26 July, 2008, pp. 59-66.
8. Solodovnikova, D.: Data Warehouse Evolution Framework. In Proceedings of the Spring Young Researcher's Colloquium On Database and Information Systems SYRCoDIS, Moscow, Russia, 2007. [online] http://ceur-ws.org/Vol-256/submission_4.pdf
9. Thalhammer, T., Schrefl, M., Mohania, M.: Active Data Warehouses: Complementing OLAP with Active Rules. Data & Knowledge Engineering, vol. 39, issue 3, December, 2001, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, pp. 241-269.
10. Garrigós, I., Gómez, J.: Modeling User Behaviour Aware WebSites with PRML. In Proceedings of the CAISE'06 Third International Workshop on Web Information Systems Modeling (WISM '06), Luxemburg, June 5-9, 2006, pp. 1087-1101.
11. F. Ravat, O. Teste: Personalization and OLAP Databases. Springer US, Annals of Information Systems, vol. 3, New Trends in Data Warehousing and Data Analysis, 2009, pp. 1-22.
12. L. Bellatreche, A. Giacometti, P. Marcel, H. Mouloudi. Personalization of MDX Queries. In Proceedings of XXIIemes journees Bases de Donnees Avancees (BDA'06), Lille, France, 2006.
13. Kimball, R., Ross, M.: The Data Warehouse Toolkit 2nd Ed: The Complete Guide to Dimensional Modeling. New York, NY: John Wiley & Sons, Inc., 2002.

14. Lenz, H.-J., Thalheim, B.: A Formal Framework of Aggregation for the OLAP-OLTP Model. Journal of Universal Computer Science, vol. 15, issue 1 (2009), pp. 273-303.
15. Inmon, W. H.: Building the Data Warehouse, 3rd ed. Wiley Computer Publishing, 2002, 428p.
16. Adamson, C.: Mastering Data Warehouse Aggregates: Solutions for Star Schema Performance. Wiley Computer Publishing, 2006, 384p.
17. Agrawal, R., Wimmers, E.: A Framework for Expressing and Combining Preferences. In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM, New York, 2000, pp. 297-306.
18. S. Borzsonyi, D. Kossmann, K. Stocker.: The Skyline Operator. In Proceedings of 17th International Conference on Data Engineering, Heidelberg, April 2001.
19. Kießling, W. Foundations of preferences in database systems. In Proceedings the International Conference on Very Large Databases (VLDB'02), Hong Kong, China, 2002, pp. 311-322 .
20. Chomicki, J.: Preference Formulas in Relational Queries. ACM TODS, 28(4), 2003, pp. 427-466.
21. Kießling, W., Köstler, G.: Preference SQL-Design, Implementation, Experiences. In Proceedings of the International Conference on Very Large Databases (VLDB'02), Hong Kong, China, 2002, pp. 990-1001.
22. P. Pu, B. Faltings, and M. Torrens. User-involved preference elicitation. In IJCAI'03 Workshop on Configuration, Acapulco, Mexico, 2003.
23. Hafenrichter, B., Kießling, W.: Optimization of Relational Preference Queries. In Proceedings of the 16th Australasian Database Conference, ADC 2005, vol. 39, Newcastle, Australia, January 31 - February 3, 2005, pp. 175-184.
24. W. Kießling: Preference Handling in Database Systems. Talk at L3S, University of Hannover, February 06, 2006.
25. Kießling, W.: Preference queries with SV-semantics. In Proceedings of COMAD'05, Goa, India, 2005, pp. 15-26.
26. Sarawagi, S.: Explaining differences in multidimensional aggregates. In Proceedings of the International Conference on Very Large Databases (VLDB'99), September 7-10, 1999, Edinburgh, Scotland, UK, pp. 42-53.
27. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. P. Brusilovsky, A. Kobsa & W. Nejdl (Eds.): The Adaptive Web (chapter 2). Springer-Verlag, Berlin, Heidelberg, 2007, LNCS 4321, pp. 54-87.
28. Kelly, D., Teevan, J.: Implicit Feedback for Inferring User Preference: A Bibliography. ACM SIGIR Forum, vol. 37, issue 2, 2003, pp. 18-28.
29. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. Kluwer Academic Publishers, USA, vol 12 , issue 4, 2002, pp. 331-370.
30. Viappiani, P., Pu, P., Faltings, B.: Acquiring User Preferences for Personal Agents. Technical Report for American Association for Artificial Intelligence (AAAI Press), 2002. [online] http://liawww.epfl.ch/Publications/Archive/Viappiani2002.pdf
31. Shearin, S., Lieberman, H.: Intelligent Profiling by Example. In: Proceedings of IUI'01, Santa Fe, New Mexico, USA, January 14-17, 2001, pp. 145-151.