

Towards Introducing User Preferences in OLAP Reporting Tool

Natalija Kozmina, Darja Solodovnikova

Faculty of Computing, University of Latvia, Raina blvd., Riga LV-1586, Latvia
{natalija.kozmina, darja.solodovnikova}@lu.lv

Abstract. This paper presents an OLAP reporting tool and an approach for determining and processing user OLAP preferences, which are useful for generating recommendations on potentially interesting reports. We discuss the metadata layers of the reporting tool including our proposed OLAP preferences metamodel, which supports various scenarios of formulating preferences of two different types: schema-specific and report-specific. The process of semantic metadata usage at the stage of formulating user preferences is also considered. The methods for processing schema-specific and report-specific OLAP preferences are outlined.

Keywords: data warehouse, user preferences, business metadata, reports

1 Introduction and Related Work

Sometimes, during sessions of work with a reporting tool, a user has no notion about what kind of data he/she is able to find there. Moreover, a user might be unaware of a potentially useful report, because, for instance, it has been created recently and the user hasn't examined it yet. In one of our works [1] we focused on acquiring user preferences implicitly either by analyzing his/her previous activities or by learning the structure of the browsed report in order to suggest him/her other reports that might be helpful, meanwhile saving user's time and effort. In this paper we concentrate on preferences explicitly formulated by users of the OLAP reporting tool.

Apart from employing the reporting tool as a means of creating, modifying and executing reports on data warehouse schema, we also consider this reporting tool as an experimental environment for introducing OLAP personalization. Users of the reporting tool may have different skill levels (e.g., expert, novice), that's why reports' recommendations based on user preferences are more valuable for novice users than for experts. The reporting tool is a part of the data warehouse framework [2] developed at the University of Latvia.

The ideas of introducing personalization into data warehouses came from the field of databases [3] and still remain a subject of interest. Data warehouse can be personalized at schema level, applying rules for the data warehouse personalization, thus, giving a user an opportunity to work with a personalized OLAP schema, which matches his/her needs [4]. Users may express their preferences on OLAP queries [5];

in such case, the problem of performing time-consuming OLAP operations to find the necessary data is significantly improved. The other method of personalizing OLAP systems is to provide query recommendations to data warehouse users via investigating former sessions of the same user [6], or via collecting user preferences into a profile and processing it, while generating query recommendations [7]. Another aspect of OLAP personalization is the visual representation of data [8]: multiple layouts and visualization techniques may be interactively used for various analysis tasks. The summary of the research made in the field of personalization in OLAP is found in one of our previous works [9].

There are some distinctive features in the approach proposed by the authors of this paper comparing to [6] and [7]. We may notice that in [6] authors analyze unexpected differences in data; however, in this paper we analyze logical structure of the reports. In [7] both data preferences and preferences on logical structure of the reports are taken into account, however, in [7] to get recommendations, user has to state his/her preferences in a user profile first. As opposed to that, in this paper no user profile is needed, because user preferences are defined automatically without asking the user to provide information directly.

In [10] a survey of the existing methods for computing data warehouse query recommendations is presented. Authors of this survey marked out four methods, which are employed to convert a certain user's query into another one that is likely to have an added value for the user: (i) methods exploiting a profile, (ii) methods based on expectations, (iii) methods exploiting query logs, and (iv) hybrid methods.

The rest of the paper is organized as follows: Section 2 introduces interrelated metadata layers of the reporting tool, i.e. logical, physical, reporting, semantic, and OLAP preferences metadata. Various user preference modeling scenarios illustrate the OLAP preference metamodel. Section 3 describes the 5-step process of user preference formulation in business language and its further transformation. Section 4 concludes the paper.

2 OLAP Reporting Tool

All operation of the data warehouse framework and the OLAP reporting tool as a part of it is based on metadata that consists of five interconnected layers (fig. 1).

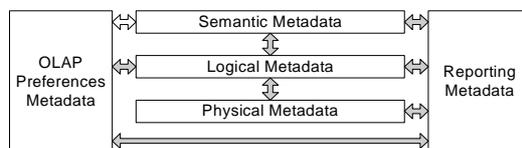


Fig. 1. Metadata connections

Logical metadata is used to describe data warehouse schemata. Physical metadata describes storage of a data warehouse in a relational database. Semantic metadata describes data stored in a data warehouse and data warehouse elements in a way that is understandable to users. Reporting metadata stores definitions of reports on data

warehouse schemata. OLAP preferences metadata stores definitions of user preferences on reports' structure and data.

Particular classes of parts of metadata are connected by associations. Semantic metadata describes report's items from the reporting metadata and data warehouse schema elements from the logical metadata. Data warehouse schema elements from the logical metadata correspond to tables and table columns described in the physical metadata. Items of reports defined in the reporting metadata are obtained from table columns described in the physical metadata and correspond to data warehouse schema elements from the logical metadata. OLAP preferences metadata defines user preferences for data warehouse schema elements described in the logical metadata and for reports described in the reporting metadata. OLAP preferences are formally defined by concepts of semantic metadata. To be more precise, components of user preferences on reports' structure are OLAP schema elements from the logical metadata that correspond to concepts from the semantic metadata, and components of user preferences on reports' data are items of reports from the reporting metadata that are defined by concepts as well. Thereby, there is a latent connection between semantic metadata and OLAP preferences metadata.

Common Warehouse Metamodel (CWM) [11] was used as a basis for the semantic, logical and physical metadata.

2.1 Logical Metadata

Metadata at the logical level describes the multidimensional data warehouse schema (fig. 2.).

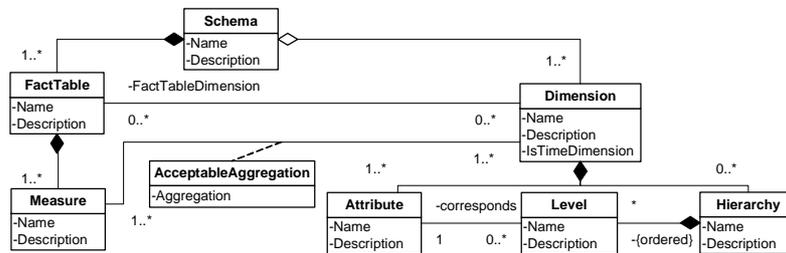


Fig. 2. Logical level metadata [12]

The logical level metadata is based on the OLAP package of Common Warehouse Metamodel (CWM) [11] and contains the main objects from this package such as dimensions with attributes and hierarchies, fact tables (cubes in CWM) with measures. Fact tables and dimensions are connected by FactTableDimension associations. OLAP package of CWM was extended by the class AcceptableAggregation, which stores information about aggregate functions (SUM, AVG, COUNT, MIN, MAX) acceptable for each measure and dimension. This metadata is essential for correct queries. The detailed description of all metadata levels of a data warehouse, including the description of the logical level, is found in the paper [12].

2.2 Reporting Metadata

Reporting metadata describes the structure of reports on data warehouse elements (fig. 3). Basically, reports are worksheets that contain data items defined by calculations, which specify computation formulas from parameters and table columns that usually correspond to schema elements (measures and attributes). Reports also consist of user-defined conditions and joins between tables.

Reports in the tool are defined by developers or experienced users themselves by choosing desired elements of a data warehouse schema and defining conditions, parameters, etc. According to the report definition, reporting metadata is created for each report. When a user runs a report in the OLAP reporting tool, an SQL query is built based on the report definition in reporting metadata [13], and its result is displayed to a user.

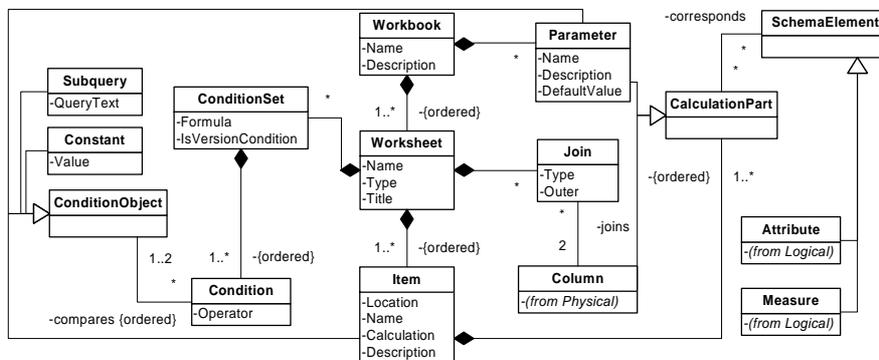


Fig. 3. Reporting metadata [13]

2.3 Semantic Metadata

It is essential for data warehouse users to understand the semantics of data that appears in reports from the business perspective.

There are multiple reasons why it is necessary to describe each element of the data warehouse model in business language. For instance, while working with the reporting tool, users also must be able to analyze this data using all necessary features, including OLAP operations drill-down and roll-up and using hierarchies. Besides, it is desirable that users can modify or construct reports themselves from elements, which are familiar to them, so that reports' creation becomes transparent. Moreover, users should be able to state their OLAP preferences, operating with business language terms, so that it would be possible to provide users of different skill levels (e.g., expert, novice) with recommendations on potentially interesting reports.

Data warehouse elements' description in business language is stored in the semantic metadata.

In CWM there is the package Business Nomenclature, which can be used to represent business metadata. This package was taken as a basis for semantic metadata depicted in fig. 4. The main classes that are used for description of data warehouse

elements are Terms and Concepts, which are united in Glossaries and Taxonomies respectively. A concept is the semantic meaning or a notion of some data warehouse element or data stored in some element, but a term is a particular word or phrase employed by users to refer to a concept. In semantic metadata Concepts define elements of a data warehouse schema (class SchemaElement from the logical metadata) and items used in reports (class Item from the reporting metadata).

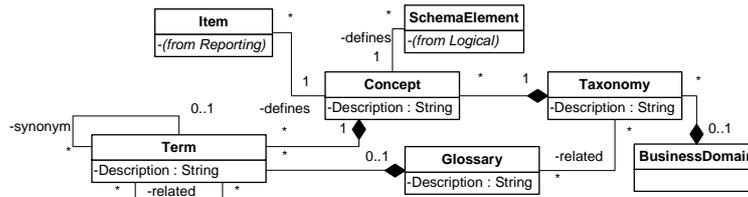


Fig. 4. Semantic metadata

2.4 OLAP Preferences Metadata

A metamodel describes OLAP schema preferences and is depicted in fig. 5. In this paper we present a revised version of the metamodel, previous versions of which are published in [14], [15].

A user may set the degree of interest (DegreeOfInterest, *DOI* [3]) for each OLAP preference. For instance, a user operates with values of the DOI attribute that may be the following: very low, low, medium, high, very high. Each DOI may have a defined real number equivalent that is assigned automatically. For example, if values of the DOI are in the interval [0; 1], then medium degree of interest corresponds to the numeric value 0.5, low degree of interest – to 0.2, etc.

In the reporting tool each workbook contains one or more worksheets, and each worksheet represents a single report. The scope of an OLAP preference may be either a specific set of reports (i.e. workbook), a single report (i.e. worksheet), or all reports defined in the reporting tool.

Each OLAP preference may be either simple (SimpleOlapPreference) or complex (ComplexOlapPreference). A complex OLAP preference consists of multiple equally important simple OLAP preferences. An advantage of a complex OLAP preference is that it allows a user to formulate sophisticated preferences assigning only one value of the degree of interest to a complex preference as a whole. For instance, *annual summary information about the average student grade in each course* is a complex OLAP preference that consists of five simple OLAP preferences (see Table 2), whereas *year=2011* is a simple OLAP preference. A simple OLAP preference may be of two types: (i) Schema-Specific preferences on OLAP schema, its elements and acceptable aggregate functions, and (ii) Report-Specific preferences on data in reports.

A PreferenceElement class describes the type of the element in user preference, which may be an OLAP schema, an OLAP schema element (e.g. dimension, fact table, attribute, measure, etc.) or a report’s item. An acceptable aggregate function

(AcceptableAggregation) may be applied to measures in order to get aggregated data w.r.t. one or many dimensions.

In report-specific preferences one or more preference elements (Items) may be included, and vice versa, a single preference element (Item) may be used in multiple user preferences of that type. Each item of the report is related to zero or one preferred term (Term) that a user selects as the most appropriate one to characterize the specific item of the report.

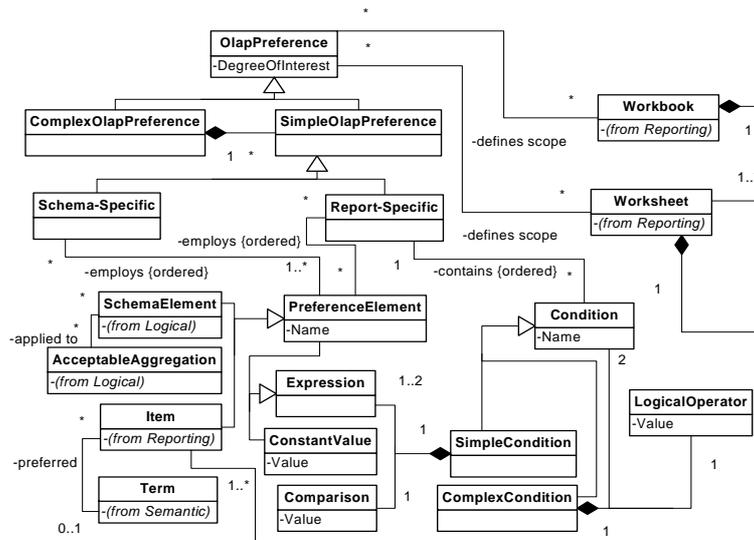


Fig. 5. OLAP preferences metadata

As report-specific preferences include restrictions on report's data, each report-specific preference may contain a set of conditions. A Condition class is divided into two subclasses: a SimpleCondition and a ComplexCondition. A complex condition consists of two or more simple conditions, joined with a logical operator (AND, OR). A simple condition consists of two expressions (Expression) and a comparison operator (Comparison). It is allowed to apply the following comparison operators: =, <>, >=, <=, >, <, in/not in, is null/is not null, like/not like, exists/not exists. Typically, one expression is a preference element and the other is a constant value (ConstantValue), which is either a string of symbols or a numeric value. There may be also just one expression, i.e. preference element, in case when the value of the comparison operator is exists/not exists or null/is not null.

We suggest several user preference modeling scenarios to motivate and illustrate the OLAP preference metamodel, demonstrated with preference examples. For more clearness, we suggest to display each example as a table with OLAP preference metamodel elements depicted as follows: (i) the simple or complex OLAP preference class is in the rightmost column, (ii) the subclasses or associated classes of either simple or complex OLAP preference are in all the rest columns, excluding the leftmost one (we intentionally did not include all classes of the metamodel in the

description of examples due to the space limitations), and (iii) instances of the most specific classes of the OLAP preferences metamodel are in the leftmost column.

Scenario A.

Description: A preference of a user contains solely an OLAP schema element or an aggregate function without any scope specified.

Type: Schema-Specific.

Scope: All worksheets in all workbooks.

In this case, a preference is schema-specific. We consider that such user preference refers to an OLAP schema element or an aggregate function, regardless of whether the given OLAP schema element or an aggregate function is used in any report or it doesn't appear in any report at all. A link to one or a set of either workbooks or worksheets is returned to the user, when user preferences are satisfied.

Example A. The user is interested in Program dimension, which contains descriptive attributes of study program. This statement could be formulated using our proposed OLAP preferences metamodel (Table 1).

Table 1. A formally described preference from the *Example A*.

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>			Simple OLAP Preference
Program	Dimension	Schema Element	Schema-Specific	
<ALL>	Workbook			
<ALL>	Worksheet			

The appearance of a certain dimension in one or several reports is not an indispensable condition. In other words, if currently there are no reports where Program dimension is involved, the preference is still retained and may be applied later, when at least one report that contains Program dimension is created.

Scenario B.

Description: A preference of a user contains an OLAP schema element or an aggregate function in the context of a certain set of reports.

Type: Schema-Specific.

Scope: One or many certain workbooks.

Apart from an OLAP schema element or an aggregate function, user states in his/her preference a certain scope of the preference. Thus, the considered preference is schema-specific with a scope set to the workbook. A link to one or a set of workbooks is returned to the user, when user preferences are satisfied.

Example B. Student Grades workbook contains multiple worksheets with reports about student exam grades, grouped by faculties, courses, years and semesters. Besides, each report has a different level of data granularity.

Assume that there are two hierarchies available – Faculty hierarchy: *Faculty* → *Course*, and Time hierarchy: *Year* → *Semester*. The user is interested in reports that represent *annual summary information about the average student grade in each course*. This preference is complex and could be split into five different preferences such as: (i) *Acceptable aggregate function is average (AVG) applied to Grades*, (ii) *Hierarchy is Faculty*, (iii) *Hierarchy level is Course*, (iv) *Hierarchy is Time*, and (v)

Hierarchy level is Year. These statements are formulated using our proposed OLAP preferences metamodel (Table 2).

Table 2. A formally described preference from the *Example B*.

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>				
AVG(Grade)	Acceptable Aggregation	Measure	Schema Element	Schema-Specific	Complex OLAP Preference
Faculty	Hierarchy				
Course	Hierarchy level				
Time	Hierarchy				
Year	Hierarchy level				
Student Grades	Workbook				
<ALL>	Worksheet				

Scenario C.

Description: A preference of a user contains restrictions on data in several reports.

Type: Report-Specific.

Scope: One or many certain workbooks.

In this scenario we point out that a preference refers to multiple reports that contain a defined value of the given item of report. We emphasize that a preference of this kind is report-specific, because it contains restrictions on certain data values of reports' items with a scope set to the workbook. A link to one or a set of workbooks is returned to the user, when user preferences are satisfied.

Table 3. A formally described preference from the *Example C*.

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>					
Program	Item			Simple Condition	Report-Specific	Complex OLAP Preference
Semester	Item	Expression				
=	Comparison					
'2011-Spring'	Constant Value	Expression				
Registrations	Workbook					
<ALL>	Worksheet					

Example C. Let's consider that a user is interested in data on students' registrations to courses during the last semester, preferably, study programs are reflected in reports. The workbook that contains reports on students' registrations is titled Registrations. So, the complex preference that will be set for the Registrations workbook is the following: *Semester item value is equal to '2011-Spring' by Program* and apparently it consists of two simple OLAP preferences, which are: (i) *Semester item value is equal to '2011-Spring'*, and (ii) *Study Program should be present in the report*. These statements could be formulated using our proposed OLAP preferences metamodel (Table 3).

Scenario D.

Description: A preference of a user contains restrictions on data in a single report.

Type: Report-Specific.

Scope: one or many certain worksheets.

In *Scenario C* a user specifies workbooks; in that way all worksheets of these workbooks are automatically included in the scope of preferences. Meanwhile, when the scope is set to worksheet (as in *Scenario D*), it signifies that a user may select arbitrary worksheets that do not necessarily belong to one and the same workbook. A link to one or a set of worksheets is returned to the user, when user preferences are satisfied. When a user runs a report of a recommended worksheet, the data in the report are already sorted in compliance with his/her preferences.

Table 4. A formally described preference from the *Example D*.

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>							
Program	Item					Report-Specific	Complex OLAP Preference	
Faculty	Item							
Program	Item	Expression	Simple Condition	Complex Condition	Report-Specific			Complex OLAP Preference
LIKE	Comparison							
'%Masters%'	Constant Value	Expression						
AND	Logical Operator							
Year	Item	Expression	Simple Condition					
=	Comparison							
'2010'	Constant Value	Expression						
Statistics	Workbook							
Graduated Students	Worksheet							

Example D. Assume that the worksheet titled Graduated Students of the Statistics workbook reflects yearly data on the total number of students that graduated in each study program. A user has stated the following complex OLAP preference that consists of three simple OLAP preferences on data of this worksheet: (i) *Study Program item should be 'Masters' of any Faculty, and Year item is set to '2010'*, (ii) *Reports with Faculties included are preferable*, and (iii) *Reports with Study Programs included are preferable*. These statements are formulated using our proposed OLAP preferences metamodel (Table 4).

Scenario E.

Description: A preference of a user contains solely restrictions on data in reports without any scope specified.

Type: Report-Specific.

Scope: all worksheets in all workbooks.

In this scenario we consider the case, when a user probably is not very familiar with the contents of workbooks or worksheets in the reporting tool (for instance, a novice user). However, he/she has a certain vision of the data that he/she would like to explore. Thus, there is a possibility for a user to set report-specific preferences with restrictions on data without specifying any scope. A link to one or a set of either workbooks or worksheets is returned to the user, when user preferences are satisfied.

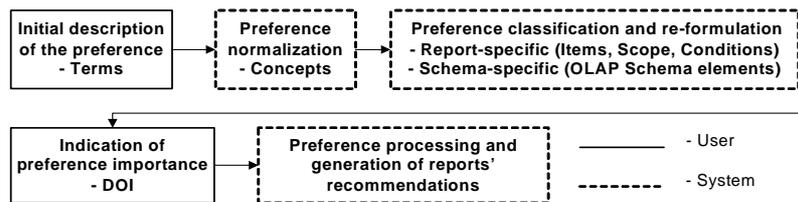
Example E. A user is looking for any reports that contain data about several courses. Say, a user states a simple OLAP preference on two courses as follows: *Course item is 'Data Warehousing' or 'IT Project Management'*. The statement is formulated using our proposed OLAP preferences metamodel (Table 5).

Table 5. A formally described preference from the *Example E*.

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>					
Course	Item	Expression	Simple Condition	Complex Condition	Report-Specific	Simple OLAP Preference
=	Comparison					
'Data Warehousing'	Constant Value	Expression				
OR	Logical Operator					
Course	Item	Expression	Simple Condition	Complex Condition	Report-Specific	Simple OLAP Preference
=	Comparison					
'IT Project Management'	Constant Value	Expression				
<ALL>	Workbook					
<ALL>	Worksheet					

3 Determining Preferences from Semantic Description

We consider semantic metadata as a means of formulating user preferences for data warehouse reports, applying pre-defined description of data warehouse elements.

**Fig. 6.** Processing user preferences described with semantic metadata

The process of preference creation and transformation is briefly depicted in fig. 6, and is the following:

Step 1 – Initial Description of the Preferences. A user describes his/her preference, choosing one of the synonym terms from the glossary that seems to be the most suitable and understandable for him/her (fig. 4.). *Example:* terms “study program”, “academic specialization”, “branch”, “field of study” are considered as synonyms, among which a user is free to select the most appropriate one. It is noteworthy that preferred terms, which the user picked out while formulating preferences, are also employed into reports to substitute the names of reports’ items, thus, making the perception of the information clearer.

Step 2 – Preference Normalization. A set of terms corresponds to exactly one concept. Thus, we normalize user preferences, transforming terms into concepts. *Example:* terms mentioned in 3.1 are all related to one concept, which is “study program”.

Step 3 – Preference Classification and Re-formulation. The type of the user preference is being detected. Bearing in mind that each concept defines either report items or OLAP schema elements, user preferences are later re-formulated, employing either items or OLAP schema elements instead of concepts (fig. 4.). If one concept corresponds to several OLAP schema elements or report items, then the number of

preferences increases respectively. The scope (worksheet or workbook) of an OLAP preference may be indicated (optionally), if the user wants to set boundaries on the analyzed reports.

(i) *Concept* \rightarrow *OLAP schema element*. If there is no data yet in a report's item or there is no report's item itself that would be linked to a concept, then a preference that contains the mentioned concept is classified as schema-specific. Such preference is reformulated so that a concept would be substituted with a corresponding OLAP schema element.

(ii) *Concept* \rightarrow *Item*. If there is any data in a report's item of a linked concept, then a preference that contains the mentioned concept is classified as report-specific and is reformulated so that a concept is substituted with a corresponding item. If it is necessary, a set of conditions may be created, employing elements from the metamodel in fig. 5. *Example*: a condition is a restriction on data, for instance, "study program name = "Information Systems"".

Step 4 – Indication of Preference Importance. In compliance with the metamodel in fig. 5, a degree of interest should be assigned by user to each OLAP preference. *Example*: a medium degree of interest is equal to 0.5 (if values of the degree of interest are normalized to the interval [0; 1]).

Step 5 – Preference Processing and Generation of Reports' Recommendations. When all OLAP preferences are formed, schema- and report-specific preferences are processed in order to provide user with recommendations on reports.

For explicitly defined schema-specific preferences, it is possible to apply the adapted hot-start method for providing recommendations on reports based on implicitly discovered schema-specific preferences described in [1]. The hot-start method is composed of two steps: firstly, user preferences for data warehouse schema elements are discovered from the history of user's interaction with the reporting tool; and secondly, we determine reports that are composed of data warehouse schema elements, which are potentially the most interesting to a user.

In case of explicitly defined preferences, the first step of the method is not applicable and must be adapted, since users specify preferences themselves. In the first step, the method should process user preferences for schema elements and propagate preferences to related schema elements. For example, if a user defines DOI for a hierarchy level, then this DOI should be propagated to the DOI of the hierarchy, which contains a level. This propagation should be proportional to the number of levels in the hierarchy.

The second step of the hot-start method should be performed, when the similarity score is calculated for each report defined in the reporting metadata and a user profile consisting of preferences. Using an adopted and adjusted formula that computes the user-item similarity score for items defined by a hierarchical ontology [16], we compute a coefficient of the hierarchical similarity between a report and a user history log. As a result, reports with the highest similarity score are recommended to a user.

However, if explicit and implicit preferences are considered together, there can be contradictory preferences, when a user explicitly defines a different DOI for a schema element then it is inferred implicitly. In such case, the explicit preference should be considered primary, so the explicitly defined DOI for a schema element should be used in the method for recommending reports.

According to the categorization of methods for computing data warehouse query recommendations presented in [10] (already mentioned in introduction and related work section of our paper), the hot-start method falls into the category of methods exploiting query logs. This method can be adapted and applied for explicitly defined schema-specific preferences.

The method for generating recommendations based on report-specific preferences is a subject of future research. However, general guidelines for such method can be specified. Firstly, reports recommended by the method based on report-specific preferences should contain items that have the highest DOI in a user profile. Secondly, recommended reports should satisfy as many conditions included into user preferences as possible.

4 Conclusions and Future Work

In this paper we paid attention to a reporting tool, developed and currently being used at the University of Latvia. We exposed five different layers of metadata that intersect each other: logical metadata that describes data warehouse schemata, physical metadata that describes storage of a data warehouse in relational database, semantic metadata that describes data stored in a data warehouse and data warehouse elements in a way that is understandable to users, reporting metadata that stores definitions of reports on data warehouse schemata, and OLAP preferences metadata that stores definitions of user preferences on reports' structure and data. We introduced various scenarios of formulating OLAP preferences as well.

We considered a possibility for a user to create OLAP preferences, using description in business language, operating with synonym terms and choosing the most appropriate among them. We briefly set forth a concept of the algorithm of OLAP preference creation, transformation and processing.

There are several directions of our future work; we would like to extend and supplement our algorithm of OLAP preference creation, transformation and processing, thus, leading it to the level, which is closer to implementation.

Also, we would like to review the existing approach for generation of reports' recommendations [1]. This approach is based on implicitly discovered schema-specific user preferences; however, it is worthwhile to adapt it to explicitly set user preferences. Along with that a method for handling report-specific user preferences should be developed. The evaluation of processing both types of explicitly set user preferences (schema- and report-specific) will follow.

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

References

1. Solodovnikova, D., Kozmina, N.: On Implicitly Discovered OLAP Schema-Specific Preferences in Reporting Tool. In: Proceedings of the 10th Int. Conf. on Perspectives in

- Business Informatics Research (BIR'11), Scientific Journal of Riga Technical University, Computer Science: Applied Computer Systems, vol. 46, pp. 35--42. (2011)
2. Solodovnikova, D.: Data Warehouse Evolution Framework. In: Proceedings of the Spring Young Researcher's Colloquium On Database and Information Systems (SYRCODIS'07), Moscow, Russia (2007), http://ceur-ws.org/Vol-256/submission_4.pdf
 3. Koutrika, G., Ioannidis, Y. E.: Personalization of queries in database systems. In Proceedings of 20th Int. Conf. on Data Engineering (ICDE'04), Boston, MA, USA, pp. 597-608. (2004)
 4. 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, pp. 401--414. (2009)
 5. Biondi, P., Golfarelli, M., Rizzi, S.: Preference-Based Datacube Analysis with MyOLAP. In Proceedings of the 27th Int. Conf. on Data Engineering (ICDE 2011), Hannover, Germany, pp. 1328--1331. (2011)
 6. Giacometti, A., Marcel, P., Negre, E., Soulet, A.: Query Recommendations for OLAP Discovery Driven Analysis. In Proceedings of 12th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP'09), Hong Kong, pp. 81--88. (2009)
 7. Jerbi, H., Ravat, F., Teste, O., Zurfluh, G.: Preference-Based Recommendations for OLAP Analysis. In Proceedings of the 11th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'09), Linz, Austria, pp. 467--478. (2009)
 8. Mansmann, S., Scholl, M. H.: Visual OLAP: A New Paradigm for Exploring Multidimensional Aggregates. In Proceedings of IADIS Int. Conf. on Computer Graphics and Visualization (MCCSIS'08), Amsterdam, The Netherlands, pp. 59--66. (2008)
 9. Kozmina, N., Niedrite, L.: Research Directions of OLAP Personalization. In: Proceedings of the 19th Int. Conf. on Information Systems Development (ISD'10), Prague, Czech Republic. (2010)
 10. Marcel, P., Negre, E.: A Survey of Query Recommendation Techniques for Data Warehouse Exploration. In 7èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA'11), vol. B-7 (Juin 2011), Clermont-Ferrand, France, pp. 119--134. (2011)
 11. Object Management Group: Common Warehouse Metamodel Specification, v1.1, <http://www.omg.org/cgi-bin/doc?formal/03-03-02>
 12. Solodovnikova, D.: Metadata to Support Data Warehouse Evolution. In: Proceedings of the 17th Int. Conf. on Information Systems Development (ISD'08), Paphos, Cyprus. pp. 627--635. (2008)
 13. Solodovnikova, D.: Building Queries on Multiple Versions of Data Warehouse. In: Haav, H.-M., Kalja, A. (eds), Databases and Information Systems V - Selected Papers from the 8th Int. Baltic Conference DBIS 2008, IOS Press, pp. 75--86. (2008)
 14. Kozmina, N., Niedrite, L.: OLAP Personalization with User-Describing Profiles. In: Forbrig, P., Günther, H. (eds.) BIR 2010. LNBIP, vol. 64, Springer, Heidelberg, pp. 188--202. (2010)
 15. Solodovnikova, D., Kozmina, N.: Determining Preferences from Semantic Metadata in OLAP Reporting Tool. In Local Proceedings of the 10th Int. Conf. on Perspectives in Business Informatics Research (BIR'11) Associated Workshops and Doctoral Consortium, Riga, Latvia, pp. 363--370. (2011)
 16. Maidel, V., Shoval, P., Shapira, B., Taieb-Maimon, M. Ontological Content-based Filtering for Personalised Newspapers: A Method and its Evaluation. Online Information Review, vol. 34 issue 5, pp. 729--756. (2010)