Baltic J. Modern Computing, Vol. 3 (2015), No. 2, 110-132

Producing Report Recommendations from Explicitly Stated User Preferences

Natalija KOZMINA

Faculty of Computing, University of Latvia, Raina blvd. 19, Riga, Latvia

natalija.kozmina@lu.lv

Abstract. There is a lack of research results on recommendations based on user preferences in data warehouse reporting tools. Moreover, report recommendations are not implemented in commercial business intelligence applications. In this paper a method for generation of report recommendations that takes explicitly stated user preferences an input is described. A user may set preferences for report elements with business terms that are later transformed into preferences on elements of the data warehouse schema (and aggregate functions) and processed. As a result, a list of report recommendations is presented to a user.

Keywords: data warehouse, reports, recommendations, user preferences

1. Introduction

In the course of time Touch-screen cellphones, laptops and other devices have become indispensable and widely used in daily life. The total amount of data is dramatically increasing from year to year, thus, leading to overload with data. For instance, according to mobile data traffic forecast by Cisco (WEB, a), the overall mobile data traffic is expected to grow to 11.2 exabytes per month by 2017, which in fact is going to be a 13-fold increase over 2012.

To accumulate large volumes of data for further analysis, data warehouses are designed and employed. "A *data warehouse* is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management decisions" (Inmon, 2002). Both desktop and web-based OLAP (*OnLine Analytical Processing*) applications are used to perform analytical tasks within a large amount of multidimensional data, which is typically stored in a data warehouse.

In business dictionary (WEB, b) *personalization* is defined as “creation of custom-tailored services that meet the individual customer’s particular needs or preferences”. Marcel (2012) gives definitions of *personalization* and *recommendation* with respect to queries. The task of *personalization* is the following: “given a database query q and some user profile compute a query q’ ⊂ q that has an added value w.r.t. the profile”. It means that given a database query q and some user profile a new query q’ enriched with preference data from the profile is constructed, moreover, query q is a part of a new query q’. The task of *recommendation* is: “given a database query q and some user profile compute a query q’ such that neither q’ ⊄ q nor q ⊄ q’ that has an added value w.r.t. the profile”. It means that given a database query q and some user profile a new query q' enriched with preference data from the profile is constructed, neither query q is a part of a new query q' nor vice versa.

Let's consider the difference between personalization and recommendation with query examples on a table that stores data about movies (i.e. title, director, genre, release year, duration, etc.). An example of the query q is: SELECT title FROM Movies WHERE director='W. Allen'. Suppose that a preference selected from the user profile is: "duration<120 min". To illustrate personalization, the query q is expanded resulting in the query q': SELECT title FROM Movies WHERE director='W. Allen' and duration<120. In its turn, the analysis of table data shows that the value 'W. Allen' for director correlates with the value 'comedy' for genre. So, in case of recommendation, the query q' would be: SELECT title FROM Movies WHERE genre='comedy' and duration<120.

1.1. Motivation

Practical experience with commercial business intelligence (BI) applications for managing data warehouse reports (e.g., Oracle Business Intelligence Discoverer (WEB, c) and MicroStrategy (WEB, d)) has shown that personalization is often limited with adjustment of visual aspect of reports and recommendations are not integrated into these tools.

A new data warehouse (or OLAP) reporting tool (see section 3) developed in the University of Latvia served as a complimentary motivation for further studies and experiments, since it is a suitable environment for implementing and testing the developed techniques of handling OLAP preferences. Users of the reporting tool may have different skill levels (e.g. expert, novice), which is why a so-called guidance based on user preferences are more valuable for novice users than for experts. The reporting tool is a part of the data warehouse framework developed at the University of Latvia.

1.2. Related Work

Report recommendation opportunities would be beneficial for business users, as they provide a valuable guidance on the exploration of the reporting tool and execution of the reports of interest. Let's say that a typical data warehouse reporting tool is the one that is designed so that logical and physical metadata conforms to CWM (Poole et al., 2003) standard. In a typical data warehouse reporting tool the emphasis is usually put on the presence of a large set of users with different experience and knowledge about data warehousing, while visualization of results plays a secondary role and often is restricted with standard graphs. Such characteristics refer to both *user session* and *user profile* analysis approaches that produce query recommendations.

The idea of recommendations with user session analysis is described in (Giacometti et al., 2009, 2011; Marcel, 2014), where OLAP server query log is examined on the subject of patterns of users' data analysis performed during previous sessions. The main point of log processing as Marcel (2014) stated is to identify the goal of user's analysis session. This can be achieved by exploring the difference between measure values in executed queries similar to those from the current session with an aim to discover a significant unexpected difference in data. Thus, all measure values are being compared, and the scope of analysis covers all sessions with the same unexpected difference in data.

An approach by Khemiri and Bentayeb (2012) helps a user construct new queries by means of analyzing user prior experience, and also falls into category of recommendations with user session analysis. This way, one can build a query using guidance as the most frequently employed query elements are detected from the past queries. The current query is linked to past ones by some association rules.

An approach that generates recommendations by means of user profile analysis is presented by Jerbi et al. (2009), and interpreted and implemented by Chaibi and Gouider (2013) to add constraints to multidimensional queries distinguishing absolute and contextual preferences and to recommend relevant queries from the log-file. A user preference stated in the user profile consists of restriction predicates on data and an analysis context that is associated with those restriction predicates. Two disjoint sets make up an analysis context: OLAP schema elements – fact tables, measures, dimensions, attributes, etc. and its values. Restrictions on measure data values (linked to an aggregate function) or conditions on attribute data values are ranked with the relevance score (a real number, [0; 1]). Preferences in the user profile with the same analysis context as in the current query are integrated into it to make the content more customized, and this query is perceived as a recommendation. If there are several recommendations generated, the system ranks them, filters out a recommendation with the highest overall score and displaying it to the user.

Aligon et al. (2014) present another direction of OLAP recommendations studies. The authors evaluate the similarity of OLAP query sequences (sessions) not queries as such, and a recommendation is a query sequence. In the latest paper of Aligon et al. (2015) collaborative filtering is added as a context for similarity calculation meaning that a set of sessions is not limited by sessions of a single user.

Neither of the report recommendation approaches with user session or user profile analysis provides recommendations on the basis of OLAP schema and its elements. The similarity of OLAP query sequences (or sessions) given in (Aligon et al., 2014, 2015) is not examined, because the "grains" compared are query sequences, not queries (or reports). An approach described in this paper (see section 2) differs from others as it produces recommendations of another kind, i.e. the likeliness on the level of logical metadata (OLAP schema, its elements, and aggregate functions) is revealed, not the likeliness in report data nor semantic terms. A method proposed in section 4 of this paper allows formulating user preferences in a way that is more understandable for a user, i.e. employing business terms. In fact, this aspect wasn't discussed in any of the approaches reviewed in this section.

2. Discovering the Most Preferred Method for Generation of Report Recommendations

In this section a quick review of three methods to support recommendations in the OLAP reporting tool and deliver data that satisfies user needs is given. These methods were empirically tested in terms of the experimentation and approved by the participants of the experimental study, which is summarized in this section.

2.1. Methods for Generation of Report Recommendations: A Summary

The methods introduced by the author of this paper generate report recommendations analyzing OLAP schema elements and are applicable for such groups of users as novice, advanced or expert. Setting preferences on logical metadata (i.e. OLAP schema elements) would help users who are unfamiliar with report structure find their data of interest.

Though the OLAP reporting tool allows users to build their own reports, nevertheless, recommendations on query construction as in (Khemiri and Bentayeb, 2012) are not considered, as it requires (i) developer rights on reports, and (ii) advanced skills, which are not necessary for a regular user, who is interested in report execution only.

The approach used for recommending reports is composed of three distinct methods – *cold-start*, *hot-start*, and *semantic hot-start*. These methods are implemented in the recommendation component of OLAP reporting tool and fall into category of the content-based filtering. Hot-start and cold-start methods for providing report recommendations involve implicitly acquired user preferences (i.e. gained automatically from user activity log), and semantic hot-start method employs user preferences stated explicitly (i.e. provided directly by the user).

Both hot-start and cold-start methods are described in details by Kozmina and Solodovnikova (2011), illustrated with real recommendation examples in (Kozmina, 2013), and briefly recalled in terms of this section. Meanwhile, the semantic hot-start method, which is the most preferred method according to the experimentation results (see section 2.2), is fully described in section 4 of this paper.

In the hot-start method user preferences for OLAP elements are detected from the user activity history stored in a log-table of the reporting tool. Then, similarity coefficients are calculated to determine the likeliness between user preferences and OLAP schema elements in each report to distinguish and recommend the potentially interesting ones. This method is best applicable for a user with rich activity history.

In the cold-start method the analysis of report structure is performed and then Vector/Cosine similarity between each report pair is calculated. Similarity coefficients are recalculated when either a new report is created, or an existing one is deleted or modified. This method is best applicable for a user with no or poor activity history. Since the report structure similarity is detected irrespective of user activity, a user is able to see any kind of reports in the list of recommendations (including newly-added and modified).

The semantic hot-start method makes use of semantic metadata is considered as a means of formulating user preferences for data warehouse reports explicitly applying business terms. A user states preferences with business terms that define OLAP schema elements and indicates a degree of interest (DOI) for each preference in a profile. The DOI is propagated to OLAP schema elements of the coarser level of granularity (i.e. dimensions, fact tables, hierarchies, schemas), and similarity between elements in user profile and each report is discovered.

To sum up, the hot-start method for providing report recommendations involves implicitly acquired user preferences, i.e. gained automatically from user activity log, so does the cold-start method, since the structure of the currently browsed report affects recommendations, while semantic hot-start method is designed for stating user preferences explicitly (i.e. setting them directly in the profile).

2.2. Experimentation Results: A Summary

The aim of the empirical study by Kozmina (2015) was to verify by means of analyzing and evaluating methods for generation of report recommendations with precision/recall technique and statistical tools, whether metadata-based report recommendations could provide valuable guidance for exploration of the OLAP reporting tool regardless of user experience and familiarity with the data. It was planned consulting the guidelines for conducting an experimental study (Kitchenham et al., 2002; Wohlin et al., 2003; Easterbrook, 2008).

One of the tasks of the practical study by Kozmina (2015) included a comparison of methods that employ user preferences gathered either explicitly or implicitly to draw conclusions on which of the two approaches was more acceptable by users, because in the papers considering report recommendations the choice of the approach to gather user preference data was not well-grounded.

The experimental study was performed in laboratory settings involving 30 subjects belonging to different groups of users (10 students, 8 academic staff representatives, and 12 administrative staff representatives) with various level of experience with reporting tools (novice - 46%, advanced users - 40%, experts - 14%). In total 70 reports on user interaction with Moodle course management system (referred as Moodle or Moodle CMS) and study process in the University of Latvia were created for an experimental study.

Each user group was assigned its own set of 4 exploratory tasks of equal complexity, whereas each task had 4 subtasks (a subtask example: "*Find a Moodle e-course category with the highest number of user sessions in the e-learning system. Session length varies from 10 to 60 min and user role is "Guest"*"). During an individual meeting each participant had to complete all the tasks, fill in the survey with 16 questions, and share their impressions in a free form. To evaluate each method and compare with others, user activity log was analyzed and direct feedback on the methods was gathered in a form of user survey and processed.

User clicks were analyzed by means of Precision, Recall and F_1-measure values to characterize how intensively the recommendation component was used. It was important to understand how exactly a user accessed and executed the report, it is, whether by means of hitting a link of the recommendation list or not.

After completing the Mann-Whitney U test on F_1-measures acquired in each mode, it was concluded that: (i) there is no significant difference in performance of the recommendation component of the reporting tool in report structure and semantic modes, (ii) meanwhile, the recommendation component in report structure or in semantic mode outperforms that in user activity mode.

According to user surveys, all tasks were mostly qualified either as easy or average. A task where the semantic hot-start was employed seemed harder than the rest, because one had to learn how to create a profile and invest their effort, but the recommendations that they received were the most precise (regardless of their skill level). All of the participants stated that report recommendations did help them solve exploratory tasks, and recommendations make the OLAP reporting tool easy-to-use and considerably speed up the process of report searching.

Summary of the user feedback showed that semantic hot-start method is more suitable for experienced users, whereas novice users prefer either cold-start as an implicit way of stating preferences or semantic mode as an explicit one. Subjects found it hard to evaluate the hot-start method in just 1 session time, although it seemed useful in everyday life, if one has to work with the same set of reports.

Regarding the type of gathering user preferences, analysis of the user activity log revealed a marginally significant difference in the performance of generating recommendations between methods that employ implicitly gathered user preferences (i.e. cold-start and hot-start) and the one that employs explicitly gathered user preferences (semantic hot-start) in favor of the latter.

3. OLAP Reporting Tool: its Architecture and Metadata

Practical experience with commercial tools for report design (e.g. Oracle Discoverer, MicroStrategy Analytics) showed that report recommendation options are missing or are introduced superficially. For example, in case of MicroStrategy, after executing a report one may select any of the “Similar reports”, however, reports are considered similar only if they belong to one and the same folder, which is created manually and does not necessarily contain reports that are alike in terms of structure or data.

The purpose of this section is to provide technical details on the implemented OLAP reporting tool and to introduce its metadata that consists of five layers, since operation of the OLAP reporting tool is based on it: logical, physical, reporting, semantic, and OLAP preferences metadata. The methods referred in section 2.1 that generate report recommendations are implemented in the OLAP reporting tool.

3.1. Architecture

The architecture of the reporting tool is composed of the server with a relational database to store data and metadata, data acquisition procedures to manage the metadata of the data warehouse schema and reports, and reporting tool components which are located on the web-server to define reports, display reports and provide recommendations on similar reports.

For the implementation of the reporting tool an Oracle database management system was used. Data acquisition procedures were implemented by means of PL/SQL procedures. The Tomcat web server was employed to allocate all the components of the reporting tool. Components that define and display reports as well as generate report recommendations are designed as Java server applets, which generate HTML code that can be used in web browsers without any extra software installation. For graphical representation of the reports an open source report engine called JasperReports was taken.

3.2. Metadata Layers

In this section five different layers of metadata of the reporting tool that intersect each other are given. Logical metadata represents data warehouse schemas and its elements, physical metadata maps data warehouse elements to a relational database, semantic metadata represents report items and data warehouse elements with terms that are comprehensible to users, reporting metadata stores data warehouse report definitions, and OLAP preferences metadata serves to express user preferences on reports and its data. CWM (Poole et al., 2003) was used as a basis for the physical, logical, and semantic metadata, and supplemented with several new classes.

Physical Metadata. CWM contains a package Relational, which was taken as a basis for physical metadata (Figure 1). It describes relational database schema of a data warehouse and the mapping of multidimensional schema elements to relational database objects. The physical metadata (Solodovnikova, 2011) is connected to the logical metadata by mappings of attributes and measures to one or several columns.

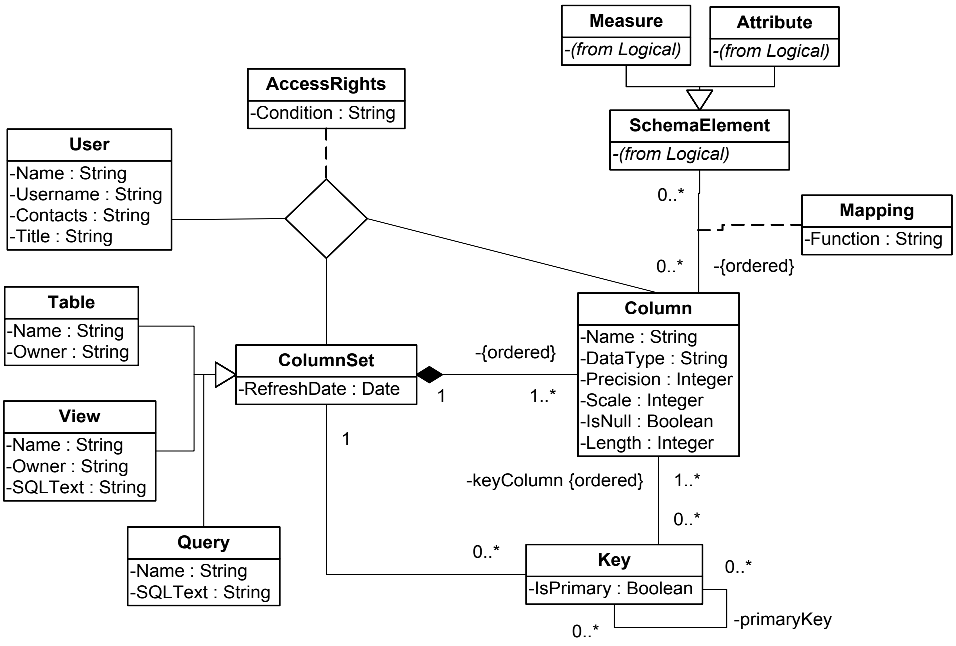


Fig. 1. Physical metadata (Solodovnikova, 2011)

Logical Metadata. Logical level metadata describes the multidimensional data warehouse schema (Figure 2) and is based on the OLAP package of CWM containing such objects as dimensions, attributes, hierarchies, fact tables (cubes in CWM), and measures (Kozmina and Solodovnikova, 2012). CWM OLAP package was extended by AcceptableAggregation class and FactTableDimension association. To link dimensions and fact tables and to be able to include them together into the same report, FactTableDimension association is required. To produce correct queries, AcceptableAggregation class stores information about aggregate functions (e.g. SUM, AVG, COUNT) acceptable for each measure and dimension.

Data warehouse schema elements (class SchemaElement) make up a hierarchical structure: a set of attributes build up a dimension, a set of measures – a fact table, while dimensions and fact tables are included into a data warehouse schema. Dimensions contain hierarchies composed of ordered levels represented by attributes.

In this paper, one takes advantage of the hierarchical structure of data warehouse schema elements to estimate the degree of interest that a user has got for schema elements located at different levels of the logical level metamodel.

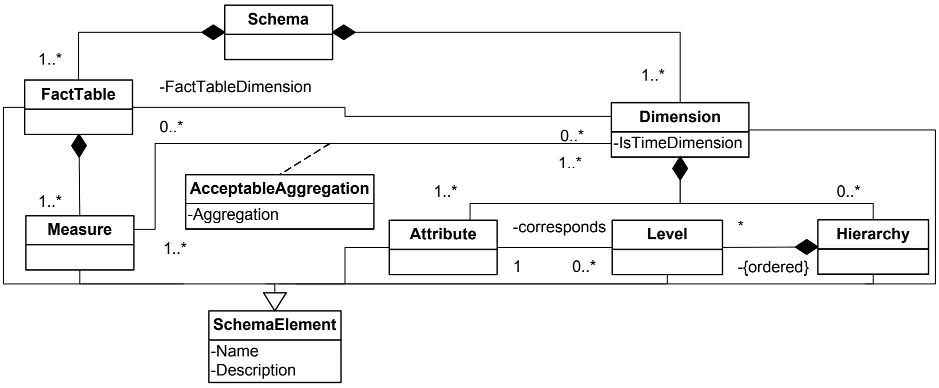


Fig 2. Logical metadata, adapted from (Kozmina and Solodovnikova, 2012)

Reporting Metadata. The structure of data warehouse reports is defined by reporting metadata (Figure 3). Although CWM contains the Information Visualization package that describes how the elements of the conceptual model of a data warehouse are displayed (e.g. as tables, cross-tabs, graphs), this metadata is insufficient. For that reason, a layer of reporting metadata was created taking as an example the visual structure of Oracle Discoverer reports.

In the reporting tool each workbook contains one or more worksheets, and each worksheet represents a single report that may contain user-defined conditions on data and table joins. Items in reports are defined by calculations (CalculationPart class) that, in fact, are formulas based on table columns that conform to such schema elements as measures or attributes and include restrictions set by parameters.

In this paper only items visible as report columns, rows, data items or page items are considered. Other items used in conditions or joins are omitted, because they are regarded as supplementary ones to the visible report items, which are interesting or useful for a user. For instance, conditions are employed to formulate restrictions on data, thus, having an impact on the contents of reports, but not on its structure. The information about item visibility is obtained from the attribute Location of the class Item in the reporting metadata.

Advanced users can define reports in the OLAP reporting tool selecting attributes and measures in terms of one data warehouse schema and setting restrictions on these elements with conditions, parameters, etc. Then, reporting metadata is generated depending on the report elements defined. When a report is executed, an SQL query is constructed on the basis of the reporting metadata.

There is a correspondence association between measures and attributes (logical metadata) and items (reporting metadata) ensured by the calculation part, thus, making logical and reporting metadata interrelated. In the reporting metadata, aggregate functions applied to measures can be detected from a formula set as a value of the Calculation attribute (Item class).

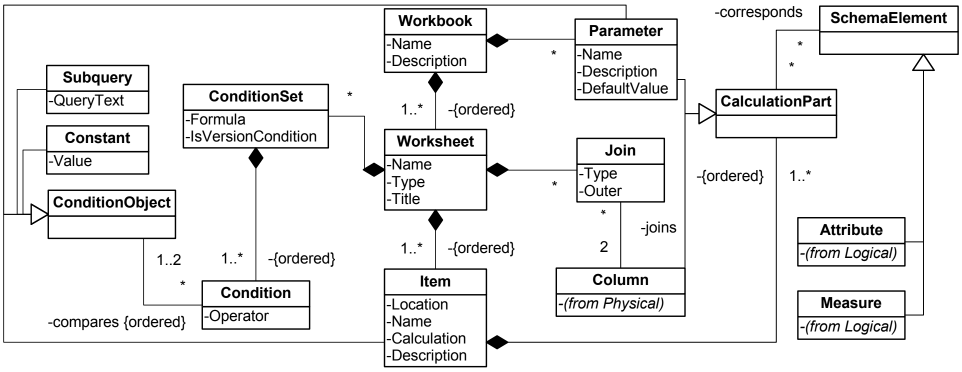


Fig. 3. Reporting metadata (Kozmina and Solodovnikova, 2012)

Semantic Metadata. In the reporting tool semantic metadata (Figure 4) serves to represent data warehouse schema elements that appear in reports in business language. Data warehouse schema elements should be mapped to some business terms for several reasons. First, one may consider adaptation of business terms that describe report items to different user groups, thus, improving user experience in report exploration and analysis by employing appropriate business terms. Then, if users have enough rights, they can construct their own report or modify the existing ones employing business terms that they are used to, this way, making report creation and modification easier. Finally, users that have different experience with reporting tools (e.g. novice, advanced user, expert) can set their own OLAP preferences with business terms, so that they could benefit from report recommendations generated on the basis of their preferences.

Business Nomenclature package from CWM contains business metadata, and it was adopted to define semantic metadata (Figure 4). A set of terms builds up a glossary and a set of concepts – a taxonomy. In semantic metadata a concept assigns a semantic meaning to some element of the data warehouse schema, i.e. attribute or measure, or some report item. Meanwhile, a term (or a set of synonym terms) is a specific word or its combination that refers to a given concept. Multiple synonym terms are of use, when one has to choose the most appropriate term out of a set while setting OLAP preferences (see Figure 7).

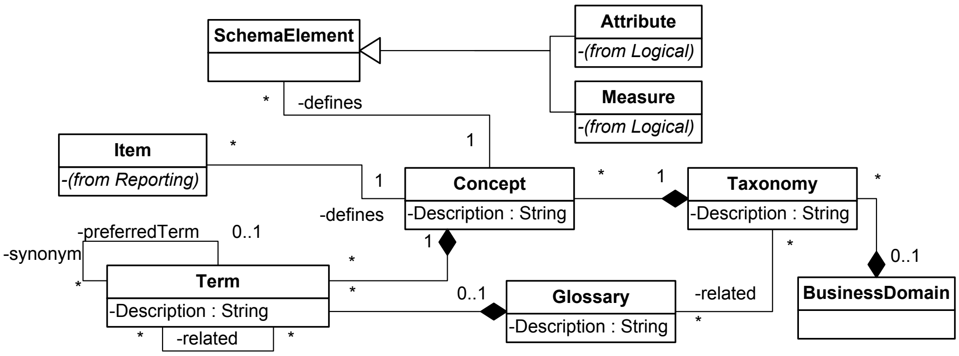


Fig. 4. Semantic metadata (Kozmina and Solodovnikova, 2012)

OLAP Preferences Metadata. A metamodel to formulate user preferences for OLAP schema elements and aggregate functions – OLAP preferences metamodel – has been proposed by Kozmina and Solodovnikova (2012) based on the empirical studies of reporting tools. OLAP preferences metadata got integrated with other metadata layers of the OLAP reporting tool, i.e. logical, physical, report, and semantic. A fragment of the complete metamodel from (Kozmina and Solodovnikova, 2012) that describes OLAP preferences is depicted in Figure 5.

A user may set the degree of interest (DegreeOfInterest, *DOI*) defined by Koutrika and Ioannidis (2004) as a real number in range [0; 1], where 0 indicates the lack of any interest, while value 1 indicates an extreme interest for each OLAP preference. Values of the DOI attribute may be either expressed qualitatively by means of categories (i.e. low, medium, etc.) or quantitatively with real numbers (i.e. low – 0.15, medium – 0.5, etc.).

An 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. In terms of this paper, the OLAP preferences that are collected and employed to generate recommendations on reports are schema-specific OLAP ones. The motivation for setting such a restriction is that methods for expressing preferences on data are put forward in studies of the other authors such as Jerbi et al. (2009), and report-specific preferences can be constructed according to the complete metamodel in (Kozmina and Solodovnikova, 2012). In its turn, the semantic hot-start method (see section 4) for processing schema-specific OLAP preferences is the original contribution of this paper.

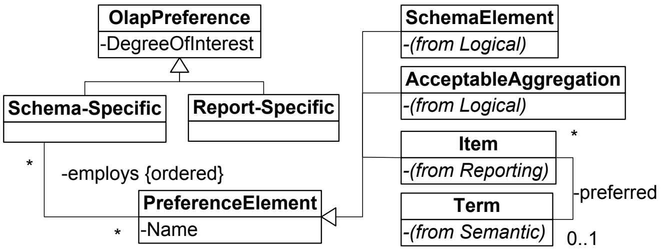


Fig. 5. A fragment of the OLAP preferences metadata

A PreferenceElement class represents each element in user preference, which may be either an OLAP schema element, a particular aggregate function, or a report item. To aggregate data, one may apply an AcceptableAggregation to measures with respect to a single or multiple dimensions. When a user sets preferences, he/she selects report items by means of preferred semantic terms that are the most suitable for the specific item of the report.

4. A Method for Giving Report Recommendations from Explicitly Stated User Preferences

A metadata-based method for construction of report recommendations that users of the OLAP reporting tool evaluated higher than other methods in the course of an experiment (see section 2.2) is presented in this section. Recommendations on potentially interesting reports are generated based on OLAP preferences that a user stated in the profile. One may set preferences manually (or explicitly) by choosing appropriate semantic terms that describe OLAP schema elements and assigning a specific degree of interest (DOI) to a particular attribute or measure, which is represented by semantic metadata (see Figure 4).

4.1. User Preferences and its Semantic Description

Semantic metadata is considered as a means of formulating user preferences for data warehouse reports explicitly applying pre-defined description of data warehouse elements. To be more precise, a user formulates his/her preferences employing terms and assigns an arbitrary degree of interest (DOI) to each preference.

Taking into consideration that terms are mapped to OLAP schema elements, the DOI of each explicitly formulated user preference is passed to the corresponding OLAP schema element of the finer level of granularity (i.e. attributes, hierarchy levels, measures) and aggregate functions. Then, the DOI is propagated to OLAP schema elements of the coarser level of granularity (i.e. dimensions, fact tables, hierarchies, schemas) – let's define this process as *upward propagation of DOI* for short.

The idea of propagating the DOI was inherited from (Guédria et al., 2007) and altered. Guédria et al. (2007) present a schema matching approach and operate with FSS (Fuzzy Subset over Schema), where a user preference degree (equivalent to DOI) is assigned to every element of a subset of elements of a schema. One of the generalization rules in (Guédria et al., 2007) says that the DOI of the element in FSS is propagated to the predecessor element preserving the same value of DOI. However, in terms of this paper the other presumption takes place – if the user shows interest in OLAP schema elements of the finer level of granularity (i.e. attributes, hierarchy levels, measures), then elements of the coarser level of granularity may also be a subject of interest for a user, though expressed to a lesser extent. This way, the DOI of the elements is not ignored and is assigned as described in *Step 5: Upward Propagation of DOI*.

4.2. Processing User Preferences Stated with Semantic Metadata

The process of preference creation and transformation is depicted in Figure 6 and is an improved version from that in (Solodovnikova and Kozmina, 2011) and (Kozmina and Solodovnikova, 2012). This section gives a consequent description of all the steps (Figure 6) to be performed.

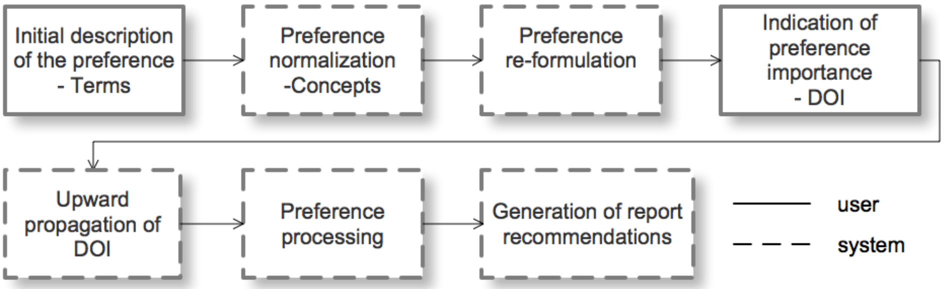
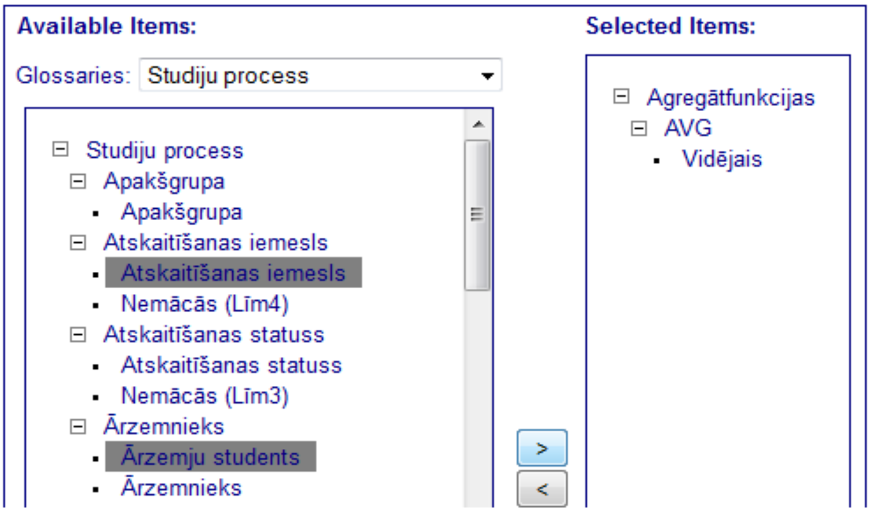

Initial description of the preference - Terms
Preference normalization -Concepts
Preference re-formulation
Indication of preference importance - DOI
Upward propagation of DOI
Preference processing
Generation of report recommendations
user
system

Fig. 6. Processing user preferences stated with semantic metadata

Step 1: Initial Description of the Preferences. OLAP schema elements are associated to items, which, in its turn, are related to terms (see OLAP preference metamodel in Figure 5). To limit the set of terms that are proposed for a user to formulate preferences, the user should select a glossary (e.g. *Study process* / Studiju process in Figure 7) that contains terms and seems to be the most suitable and understandable for him/her (see Figure 4). Next, a user describes his/her preference choosing one of the synonym terms (e.g. *Expulsion reason* / Atskaitīšanas iemesls and *Foreign student* / Ārzemju students in Figure 7) from the glossary.

Example. Terms "foreign student" and "foreigner" in the context of study process are considered synonyms, from which a user is free to select the most appropriate one.



Glossaries: Studiju process

Fig. 7. The process of choosing semantic terms in the reporting tool

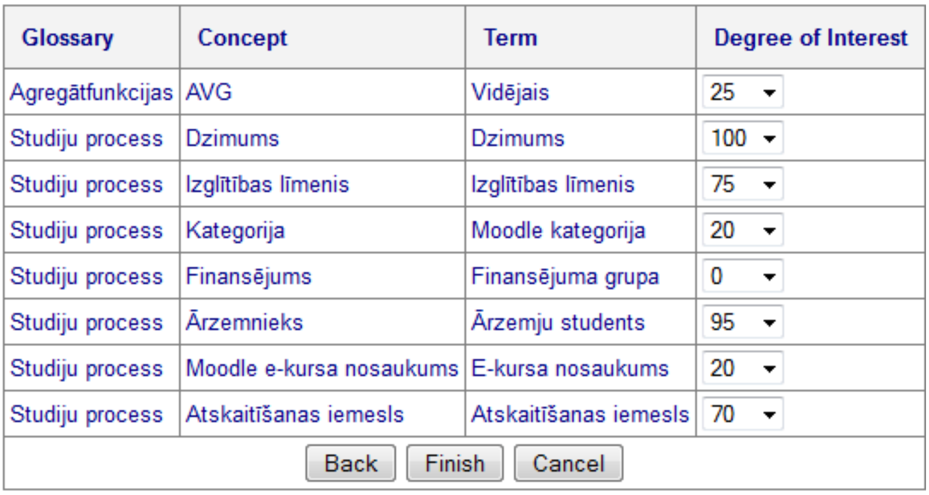
Step 2: Preference Normalization. A set of terms corresponds to exactly one concept (see Figure 4). Thus, user preferences are normalized transforming terms into concepts.

Example. Terms "study program", "academic specialization", "branch", "field of study" are all related to one concept, which is "study program".

Step 3: Preference Re-formulation. Knowing that each concept defines OLAP schema elements (see Figure 4) user preferences are re-formulated employing OLAP schema elements instead of concepts. If one concept corresponds to several schema elements, then the number of preferences increases respectively.

Step 4: Indication of Preference Importance. In compliance with the metamodel in Figure 5, a user should assign a DOI to each of the OLAP preferences.

Example. Values of the degree of interest are normalized to the interval [0; 1]. To ease the perception of DOI coefficient values, for instance, the values may be split into several intervals that characterize the DOI: very low [0; 0.2], low (0.2; 0.4], average (0.4; 0.6], high (0.6; 0.8], and very high (0.8; 1]; or displayed as natural numbers from 1 to 100 as shown in Figure 8, thus, providing a typical numerical scale for assessment of the DOI. Quantitative values of the DOI are employed for further processing of preferences.



Glossary	Concept	Term	Degree of Interest
Agregātfunkcijas	AVG	Vidējais	25
Studiju process	Dzimums	Dzimums	100
Studiju process	Izglītības līmenis	Izglītības līmenis	75
Studiju process	Kategorija	Moodle kategorija	20
Studiju process	Finansējums	Finansējuma grupa	0
Studiju process	Ārzemnieks	Ārzemju students	95
Studiju process	Moodle e-kursa nosaukums	E-kursa nosaukums	20
Studiju process	Atskaitīšanas iemesls	Atskaitīšanas iemesls	70

Fig. 8. The process of setting degrees of interest for chosen elements

Step 5: Upward Propagation of DOI. When a user runs a report, attributes and measures used in the report are obtained by means of the reporting metadata analysis. After the schema elements used in the report are determined, user's degree of interest for all employed schema elements is updated hierarchically starting from the elements of the finer level of granularity.

Algorithm 1 provides upward propagation of the DOI and is executed for each attribute or measure that has a corresponding DOI defined by user in the profile by means of semantic metadata. For any other attribute or measure that is not derivable from user preferences stated in the profile the DOI is equal to 0. In this algorithm, the degree of interest for elements of the finer level of granularity is propagated to elements of the coarser level proportionally to the total number of finer level elements belonging to each element of coarser level of granularity.

Algorithm 1.

Input: Explicitly set user OLAP preferences for schema elements with the degrees of interest set for OLAP schema element *E* derived from semantic metadata in user profile. DOI(*SE*) is the user's degree of interest for the schema element *SE* derived from semantic metadata in user profile or calculated using the upward propagation of the DOI.

Output: User OLAP preferences with updated degrees of interest

```
factTables = ∅; // a set of fact tables related to E, if E is a measure
dimensions = ∅;  // a set of dimensions related to E, if E is an attribute

foreach E in E.first..E.last loop
    // if element E is a measure
    if E instanceOf(Measure) then
        // getting a fact table, which the measure E belongs to
        F=getFactTable(E);
        DOI(F)=DOI(F)+DOI(E)/countMeasures(F);
        if F not in factTables then
            add(F, factTables);
        end if;
    // if element E is an attribute
      else if E instanceOf(Attribute) then
        // getting a dimension, which the attribute E belongs to
        D=getDimension(E);
        DOI(D)=DOI(D)+DOI(E)/countAttributes(D);
        if D not in dimensions then
            add(D, dimensions);
        end if;
         // getting hierarchies, levels of which correspond to the attribute E
        hierarchies=getHierarchies(E);
        foreach H in hierarchies do
            DOI(H)=DOI(H)+DOI(E)/countLevels(H);
        end loop;
    end if;
end loop;
foreach F in factTables loop
// getting a schema, which the fact table F belongs to
  S=getSchema(F);
  DOI(S)=DOI(S)+DOI(F)/countFactTables(S);
end loop;
foreach D in dimensions loop
  // getting schemas, which the dimension D belongs to
  schemas=getSchemas(D);
    foreach S in schemas loop
        DOI(S)=DOI(S)+DOI(D)/countDimensions(S);
     end loop;
end loop;
```

The degree of interest *DOI(E$_i$)* is a value stated by user in the profile manually and normalized to [0..1]; E_i is an OLAP schema element of the finer level of granularity, i.e. an attribute referred as A_i or a measure referred as M_i. If some attribute turns out to be a level of a hierarchy, then this level is also assigned the same DOI. For any other E_i that are not derivable from user preferences stated in the profile the *DOI(E$_i$)* is equal to 0.

If the element is a measure M_j, then the degree of interest of a fact table F_i is defined as shown in Formula 1:

$$DOI(F_i) = \sum_{j=1}^{k} \frac{DOI(M_j)}{n}, \qquad (1)$$

where *DOI(M$_j$)* are the values of the DOI of measures belonging to a fact table F_i that were detected from user profile preferences, k is the total number of measures belonging to a fact table F_i that were detected from user profile preferences, and n is the total number of measures in a fact table F_i.

If the element is an attribute A_j, then the DOI of a dimension D_i is defined as shown in Formula 2:

$$DOI(D_i) = \sum_{j=1}^{k} \frac{DOI(A_j)}{n}, \qquad (2)$$

where $DOI(A_j)$ are the values of the DOI of attributes belonging to a dimension D_i that were detected from user profile preferences, k is the total number of attributes belonging to a dimension D_i that were detected from user profile preferences, and n is the total number of attributes in a dimension D_i.

The degree of interest of a hierarchy H_i is defined as shown in Formula 3:

$$DOI(H_i) = \sum_{j=1}^{k} \frac{DOI(A_j, D_l)}{n}, \qquad (3)$$

where D_l is the dimension, which the attribute A_j belongs to, $DOI(A_j, D_l)$ are the values of the DOI of attributes detected from user profile preferences belonging to a dimension D_l, which, in fact, are levels of hierarchy H_i, k is the total number of attributes detected from user profile preferences that are levels of hierarchy H_i, and n is the total number of levels in a hierarchy H_i.

Finally, the degree of interest of a schema S_i is defined as shown in Formula 4:

$$DOI(S_i) = \sum_{j=1}^{k} \frac{DOI(D_j)}{d} + \sum_{l=1}^{m} \frac{DOI(F_l)}{f}, \qquad (4)$$

where $DOI(D_j)$ are the values of DOI of dimensions belonging to a schema S_i that were detected from user profile preferences, k is the total number of dimensions belonging to a schema S_i that were detected from user profile preferences, d is the total number of dimensions in a schema S_i, $DOI(F_l)$ are the values of DOI of fact tables belonging to a schema S_i that were detected from user profile preferences, m is the total number of fact tables belonging to a schema S_i that were detected from user profile preferences, and f is the total number of fact tables in a schema S_i.

Also, a user may state in the profile the DOI for aggregate functions. After updating the degrees of interest for schema elements, the degrees of interest of all acceptable aggregations used in the report are updated. For each triple of measure, attribute, and aggregate function applied to the measure the acceptable aggregation is obtained, and its degree of interest is increased by the same value that was stated by a user in the profile.

Note that the degrees of interest are only calculated for the current preference elements in user profile. For instance, if at first a user stated a preference P1: “Study Program, DOI = 0.9 (very high)” and afterwards replaced it with P2: “Faculty, DOI = 0.6 (average)”, then in newly-generated recommendations only P2 will be taken into account and all the degrees of interest calculated by upward propagation of DOI for P1 (since Program and Faculty are levels of the same hierarchy as depicted in Figure 10) will be deleted.

Step 6: Preference Processing. When all OLAP preferences are formed and DOI assigned, they are processed in order to provide user with recommendations on reports.

In case of explicitly defined preferences, the similarity score is calculated for each report defined in the reporting metadata and a user profile consisting of preferences. To calculate the similarity score between a report and a user profile, the *hierarchical similarity* between a report and a user profile is computed as shown in Formula 5:

$$sim = \frac{\sum_{i=1}^{n} DOI(E_i)}{\sum_{j=1}^{m} DOI(P_j)}, \tag{5}$$

where $E_1,...,E_n$ are schema elements used in the report, and $P_1,...,P_m$ are all schema elements derived from semantic description defined by user in the profile.

Formula 5 was the result of considering the formula employed to compute the user-item similarity score for items defined by a hierarchical ontology by Maidel et al. (2010) in the context of the OLAP schema elements. Maidel et al. (2010) deal with hierarchical ontology of news, they collect concepts (e.g., "life style", "politics", "crime", "elections") that a user is interested in into user profile and compare them to concepts in items, i.e. newspapers. The user-item similarity score is computed as a ratio of the number of hits on the set of concepts in an item's profile multiplied with the score of similarity to the number of hits on the set of concepts in a user's profile. The score of similarity in this case is a real number from 0 ("no match at all") to 1 ("perfect match").

Step 7: Generation of Report Recommendations. In practice, there are two types of similarity coefficient calculated: *fact-based* (i.e. value of hierarchical similarity is calculated for each report for measures, fact tables, and schemas) and *dimension-based* (i.e. for attributes, hierarchies, dimensions, and schemas). It has been decided to distinguish two types of similarity coefficients due to the well-known characteristics of the data stored in data warehouses, i.e. quantifying (measures) and qualifying (attributes).

The essence of any data warehouse is in facts, while the describing attributes give the auxiliary information, however, practical experience shows that attributes make reports differ from one another while facts remain the same (in terms of one OLAP schema). Thereby, the recommendations are filtered (i) firstly, by the value of *fact-based* similarity coefficient, (ii) secondly, by the value of the *dimension-based* similarity coefficient, and (iii) finally, by aggregate function DOI.

An Example of Calculating Similarity Values Based of User Preferences (Step 1-7). To demonstrate the semantic hot-start method for recommending OLAP reports, let's consider an example of a data warehouse schema, which stores data about students.

The logical metamodel of the example schema *Students* (Figure 9) consists of two fact tables: *Registrations* and *Enrolment*, and four dimensions: *Time*, *Program*, *Status*, and *Course*. *Registrations* fact table stores information about the number of students, registered for studies at the university per study program (dimension *Program*) and date (dimension *Time*). *Enrolment* fact table contains data about the number of students, enrolled into courses, the number of enrolments and its cancellations for each course (dimension *Course*), study program (dimension *Program*), status (dimension *Status*), and date (dimension *Time*).

Dimensions *Time* and *Program* build up hierarchies with corresponding levels as shown in Figure 10.

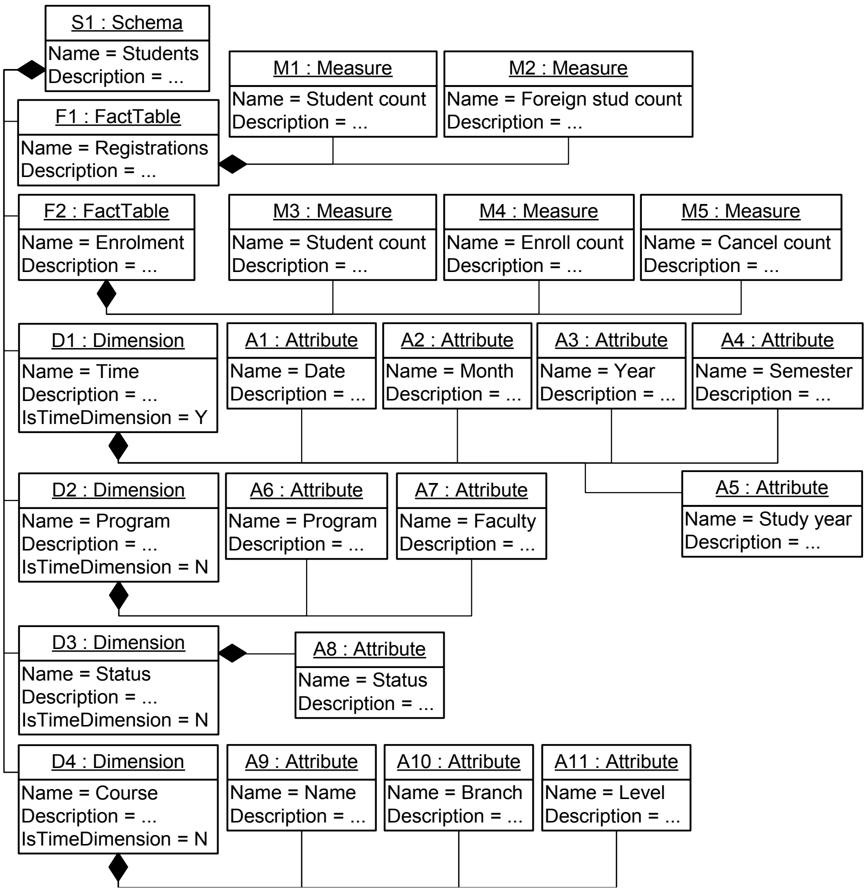


Fig. 9. *Students* data warehouse schema (Kozmina and Solodovnikova, 2011)

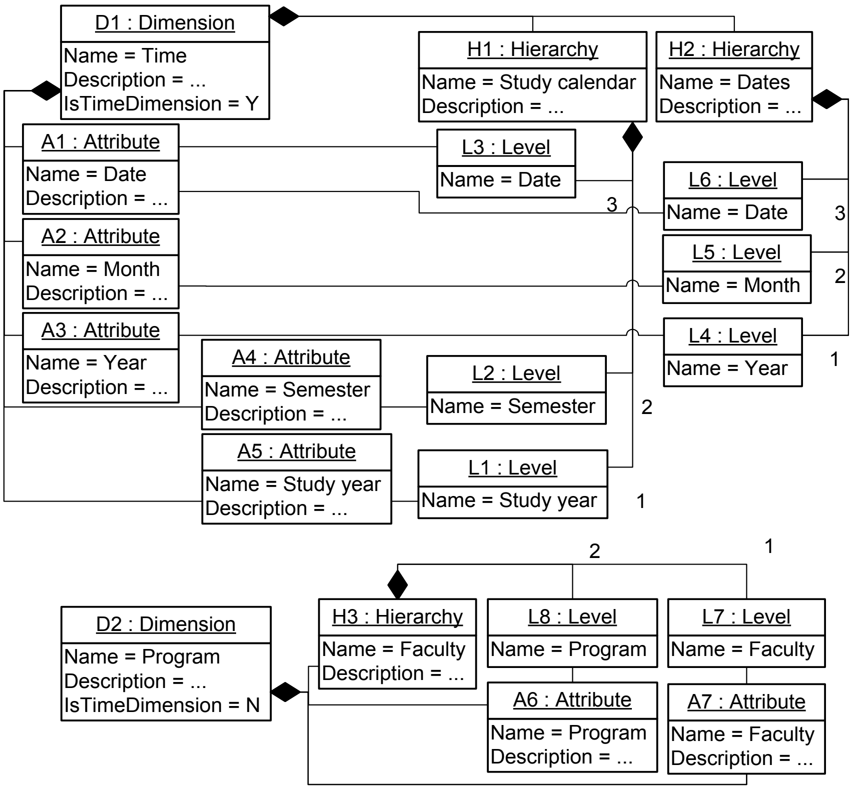


Fig. 10. Hierarchies of the dimensions *Time* and *Program* (Kozmina and Solodovnikova, 2011)

Suppose that a user set arbitrary preferences with semantic terms, which all refer to a glossary *Study process*. In total there are 9 explicitly set preferences with a degree of interest assigned to each of them. Mapping of semantic metadata elements (terms and concepts) and corresponding data warehouse schema elements (either attributes or measures), as well as aggregate function values and DOI values are proposed in Table 1.

Preferences P1-P8 are mapped to elements of *Students* schema (Figure 9), whereas P9 refers to *Gradebook* schema. The calculation of hierarchical similarity values by means of semantic hot-start method will be illustrated with two reports: R_1 – *Average foreign student count for each study program per semester*, and R_2 – *Total student count enrolled into courses for each faculty per year*. Both of the reports contain elements from schema *Students*, which is why in this particular example preference P9 will have no effect on hierarchical similarity values as its corresponding schema element (measure Average student grade) refers to schema *Gradebook*. Thus, P9 will be omitted.

Employing an Algorithm 1 for propagation of DOI from the elements of the finer level of granularity (attributes and measures; see Table 1), the DOI values for the elements of the coarser level of granularity (dimensions, fact tables, hierarchies, schema; see Figure 9 and 10) are computed. In Table 2 the values of user's degree of interest (DOI) for all attributes, dimensions, fact tables, and a schema itself are shown.

Table 1. Mapping of semantic metadata elements and data warehouse schema elements

	Semantic Metadata Element		***Schema Element***		***Aggr. Function***	***DOI***
	Term	***Concept***	***Attribute***	***Measure***		
P1	Academic specialization	Study program	A6: Program	-	-	0.75 (high)
P2	Faculty	Faculty	A7: Faculty	-	-	0.9 (very high)
P3	Number of enrolled students	Student count	-	M3: Student count	-	0.5 (average)
P4	Year	Year	A3: Year	-	-	1 (very high)
P5	Number of foreign students	Foreign student count	-	M2: Foreign stud count	-	0.4 (low)
P6	-	-	-	-	SUM	0.85 (very high)
P7	-	-	-	-	AVG	0.35 (low)
P8	Course title	Course	A9: Name	-	-	0.55 (average)
P9	Average student grade	Average student grade	-	M6: Average stud grade	-	0.7 (high)

Table 2. DOI values of Students data warehouse schema and its elements

	Schema	***Fact Tables***		***Measures***					***Dimensions***			
	S_1	F_1	F_2	M_1	M_2	M_3	M_4	M_5	D_1	D_2	D_3	D_4
DOI	$\frac{233}{480}$	$\frac{2}{10}$	$\frac{1}{6}$	0	$\frac{4}{10}$	$\frac{1}{2}$	0	0	$\frac{1}{5}$	$\frac{33}{40}$	0	$\frac{11}{60}$

	Attributes										
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
DOI	0	0	1	0	0	$\frac{3}{4}$	$\frac{9}{10}$	0	$\frac{11}{20}$	0	0

The values of user's degree of interest (DOI) for hierarchies with levels composed of attributes are such as shown in Table 3.

Table 3. DOI values of hierarchy levels and hierarchies of Students data warehouse schema

	Hierarchies			***Attributes/Hierarchy Levels***							
				Hierarchy H_1			***Hierarchy H_2***			***Hierarchy H_3***	
	H_1	H_2	H_3	A_5	A_4	A_1	A_3	A_2	A_1	A_7	A_6
DOI	0	$\frac{1}{3}$	$\frac{33}{40}$	0	0	0	1	0	0	$\frac{9}{10}$	$\frac{3}{4}$

The hierarchical similarity values for the reports R_1 and R_2 are computed separately for fact-based recommendations $simF_{R1}$ (Formula 6) and $simF_{R2}$ (Formula 6), and for dimension-based recommendations $simD_{R1}$ (Formula 8) and $simD_{R2}$ (Formula 9) respectively.

For short, let's substitute the sum of all schema elements detected from user preferences profile with *DOI(p)*, where: $DOI(p) = DOI(S_1) + DOI(F_1) + DOI(F_2) + DOI(M_2) + DOI(M_3) + DOI(D_1) + DOI(D_2) + DOI(D_4) + DOI(A_3) + DOI(A_6) + DOI(A_7) + DOI(A_9) + DOI(H_2) + DOI(H_3) \approx 6.31875$.

$$simF_{R1} = \frac{DOI(M_2) + DOI(F_1) + DOI(S_1)}{DOI(p)} \approx \frac{1.085}{6.31875} \approx 0.17 \tag{6}$$

$$simF_{R2} = \frac{DOI(M_3) + DOI(F_2) + DOI(S_1)}{DOI(p)} \approx \frac{1.152}{6.31875} \approx 0.18 \tag{7}$$

$$\begin{aligned} simD_{R1} &= \frac{DOI(S_1) + DOI(D_2) + DOI(A_6) + DOI(H_3)}{DOI(p)} + \\ &+ \frac{DOI(D_1) + DOI(A_4) + DOI(H_1)}{DOI(p)} \approx \frac{3.085}{6.31875} \approx 0.49 \end{aligned} \tag{8}$$

$$\begin{aligned} simD_{R2} &= \frac{DOI(S_1) + DOI(D_2) + DOI(A_7) + DOI(H_3)}{DOI(p)} + \\ &+ \frac{DOI(D_1) + DOI(A_3) + DOI(H_2)}{DOI(p)} \approx \frac{4.568}{6.31875} \approx 0.72 \end{aligned} \tag{9}$$

According to the fact-based similarity values between the OLAP preferences in the user profile and the reports R_1 and R_2, the report R_2 is ranked higher than the report R_1. In compliance with the dimension-based similarity values, the reports are ordered the same way, whereas similarity coefficient value of R_2 significantly exceeds that of R_1.

5. Conclusions and Discussion

Gauch et al., 2007 state that there is much controversy in explicit methods for gathering user preferences. For example, on one hand, the results of its performance are quite precise, because the preferences are set by a person and are not calculated employing any kind of implicit methods to analyze user data; on the other hand, the user is not always willing to express the preferences as he/she is not motivated enough. However, the results of an empirical study performed by Kozmina (2015) on a set of subjects (which had different skill level with reporting tools) that compared 2 implicit (cold-start and hot-start) methods and 1 explicit (semantic hot-start) method of stating and processing metadata-based user preferences proved the following. Even though the users had to invest some extra effort in developing their profiles explicitly, the recommendations generated by the semantic hot-start method were the most precise.

The main contribution of this paper is a detailed description of the semantic hot-start method. The semantic hot-start method converts user preferences that are explicitly set with appropriate business terms to preferences on OLAP schema elements and aggregate functions. Steps to be performed (i) to state user preferences in a profile, (ii) to assign a degree of interest (DOI) that is propagated to OLAP schema elements of the coarser level of granularity with a specific algorithm, and then (iii) to process them are explained systematically. The final outcome of the semantic hot-start method is a list of report recommendations (links).

As to future work, the ability to rate recommendations (e.g. likes/dislikes, "stars") could add value for the calculation of similarity values in report recommendation methods and improve the quality of recommendations. Besides, in the long term, the recommendation component (which includes cold-start, hot-start, and semantic hot-start methods) may be parametrized. This way, it could be integrated into other reporting tools that support multidimensional structure and have physical, logical, and semantic level metadata in compliance with CWM standard (Poole et al., 2003) to take advantage of user preferences.

References

Aligon J., Golfarelli M., Marcel P., Rizzi S., Turricchia E. (2014) Similarity Measures for OLAP Sessions. *Knowledge and Information Systems*, 39(2): 463-489

Aligon J., Gallinucci E., Golfarelli M., Marcel P., Rizzi S. (2015) A Collaborative Filtering Approach for Recommending OLAP Sessions. *Decision Support Systems*, 69(2015):20-30

Chaibi, N., Gouider, M.S. (2013) Personalization and Recommendation of Queries in Multidimensional Data Base. *International Journal of Engineering Science Invention (IJESI)*, 2(5):74-80.

Easterbrook, S., Singer, J., Storey, M.A., Damian, D. (2008) Selecting Empirical Methods for Software Engineering Research. Shull, F. et al. (eds.): *Guide to Advanced Empirical Software Engineering*. Springer-Verlag, London, pp. 285-311.

Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A. (2007) User Profiles for Personalized Information Access. Brusilovsky, P. et al. (eds.): *The Adaptive Web (chapter 2)*. Springer-Verlag, Berlin, Heidelberg, LNCS 4321, pp. 54-87.

Giacometti, A., Marcel, P., Negre, E., Soulet, A. (2009) 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.

Giacometti, A., Marcel, P., Negre, E., Soulet, A. (2011) Query Recommendations for OLAP Discovery-driven Analysis. *Data Warehouse Mining*, 2011, 7(2):1-25.

Guédria, W., Bellahsene, Z., Roche, M. (2007) A Flexible Approach Based on the user Preferences for Schema Matching. *In Proceedings of the 1st International Conference on Research Challenges in Information Science (RCIS'07)*, Ouarzazate, Morocco, pp. 21-26.

Inmon, W. H. (2002) *Building the Data Warehouse, 3rd ed.* Wiley Computer Publishing, 428p.

Jerbi, H., Ravat, F., Teste, O., Zurfluh, G. (2009) Preference-Based Recommendations for OLAP Analysis. *In Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'09)*, Linz, Austria, pp. 467-478.

Khemiri, R., Bentayeb, F. (2012) Interactive Query Recommendation Assistant. *In Proceedings of the 23rd International Workshop on Database and Expert Systems Applications (DEXA'12)*, IEEE, 2012, Vienna, Austria, pp. 93-97.

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., Rosenberg, J. (2002) Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Transactions on Software Engineering*, 28(8): 721-734.

Kozmina, N. (2015) An Empirical Study of Recommendations in OLAP Reporting Tool. *In Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS'15)*, Barcelona, Spain, vol. 1, pp. 303-312.

Kozmina, N. (2013) Adding Recommendations to OLAP Reporting Tool. *In Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS'13)*, Angers, France, vol. 1, pp. 238-245.

Kozmina, N., Solodovnikova, D. (2011) On Implicitly Discovered OLAP Schema-Specific Preferences in Reporting Tool. *In Proceedings of the 10th International Conference on Perspectives in Business Informatics Research (BIR'11)*, Riga, Latvia. Scientific Journal of Riga Technical University, Computer Science: Applied Computer Systems, 46:35-42.

Kozmina, N., Solodovnikova, D. (2012) Towards Introducing User Preferences in OLAP Reporting Tool. In: Niedrite, L., et al. (eds.) *BIR 2011 Workshops*, Riga, Latvia. Springer, Heidelberg, LNBIP, vol. 106, pp. 209-222.

Koutrika, G., Ioannidis, Y. E. (2004) Personalization of Queries in Database Systems. *In Proceedings of the 20th International Conference on Data Engineering (ICDE'04)*, Boston, MA, USA, March 30 – April 2, pp. 597-608.

Maidel, V., Shoval, P., Shapira, B., Taieb-Maimon, M. (2010) Ontological Content-based Filtering for Personalised Newspapers: A Method and its Evaluation. *Online Information Review*, vol. 34 issue 5, pp. 729-756.

Marcel, P. (2012) OLAP Query Personalisation and Recommendation: An Introduction. In: Aufaure, M.-A. and Zimányi, E. (eds.) *eBISS 2011*, Springer, Heidelberg, LNBIP, vol. 96, pp. 63-83.

Marcel, P. (2014) Log-driven User-centric OLAP. *In Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'2014)*, Opatija, Croatia, IEEE, 2014, pp. 1446-1451.

Poole J., Chang D., Tolbert D., Mellor D. 2003. *Common Warehouse Metamodel Developers Guide*. Wiley Publishing, 704p

Solodovnikova, D. (2011) Evolution-Oriented User-Centric Data Warehouse. *In Proceedings of the 19th International Conference on Information Systems Development (ISD'10)*, Springer Science+Business Media, pp. 721-734.

Solodovnikova, D., Kozmina, N. (2011) Determining Preferences from Semantic Metadata in OLAP Reporting Tool. *In Local Proceedings of the 10th International Conference on Perspectives in Business Informatics Research (BIR'11)*, Associated Workshops and Doctoral Consortium, Riga, Latvia, pp. 363-370.

Wohlin, C., Höst, M., Henningsson, K. (2003) Empirical Research Methods in Software Engineering. In: A.I. Wang, A.I. and Conradi, R. (eds.) *Experiences from ESERNET*, LNCS, Springer Verlag, pp. 7-23.

WEB (a) Cisco report http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

WEB (b) Business Dictionary http://www.businessdictionary.com/definition/personalization.html

WEB (c) Oracle Business Intelligence Discoverer http://www.oracle.com/technetwork/developer-tools/discoverer/overview/index.html

WEB (d) MicroStrategy http://www.microstrategy.com/software/products/report-services

Received June 1, 2015, revised June 10, 2015, accepted June 11, 2015