

UNIVERSITY OF LATVIA

NATĀLIJA KOZMINA

**METADATA-BASED PERSONALIZATION  
IN DATA WAREHOUSES**

Doctoral thesis for Ph. D. (Dr. sc. comp.) academic degree

Field: computer science

Subfield: data processing systems and computer networks

Advisor:

Asoc. professor, Dr. sc. comp.

LAILA NIEDRĪTE

R ī g a - 2014

## CONTENTS

<b>1. INTRODUCTION .....</b>	<b>4</b>
<b>1.1. Motivation, Topicality and Novelty of the Subject .....</b>	<b>6</b>
1.1.1. Motivation .....	6
1.1.2. Topicality and Novelty .....	7
<b>1.2. Goals and Tasks of the Thesis .....</b>	<b>8</b>
<b>1.3. Hypotheses Formulated in the Research .....</b>	<b>9</b>
<b>1.4. Research Methods Applied .....</b>	<b>10</b>
<b>1.5. Main Results of the Research.....</b>	<b>10</b>
<b>1.6. Approbation of the Results .....</b>	<b>12</b>
<b>1.7. Outline of the Thesis .....</b>	<b>14</b>
<b>2. LITERATURE REVIEW ON DATA WAREHOUSE PERSONALIZATION .....</b>	<b>16</b>
<b>2.1. The Intent of the Section .....</b>	<b>16</b>
<b>2.2. Research Directions in OLAP Personalization .....</b>	<b>16</b>
2.2.1. OLAP Schema, its Elements and Basic OLAP Operations .....	16
2.2.2. A Description of OLAP Personalization Directions .....	18
2.2.3. A Comparison of Existing OLAP Personalization Approaches .....	23
2.2.4. Hard and Soft Constraints as User Preferences.....	25
2.2.5. Approaches for Collecting User Preference Data .....	28
2.2.6. Methods for Obtaining User Preferences.....	28
<b>2.3. Summary of the Section.....</b>	<b>30</b>
<b>3. REQUIREMENT FORMALIZATION TO DEVELOP THE CONCEPTUAL MODEL OF A DATA WAREHOUSE IN COMPLIANCE WITH USER NEEDS .....</b>	<b>33</b>
<b>3.1. The Intent of the Section .....</b>	<b>33</b>
<b>3.2. Methods to Construct Conceptual Models for Data Warehouses .....</b>	<b>33</b>
<b>3.3. Existing Methods for Formalization of Data Warehouse Requirements .....</b>	<b>35</b>
<b>3.4. Requirement Formalization Metamodel and Examples .....</b>	<b>36</b>
3.4.1. Principles of Requirement Reformulation .....	37
3.4.2. Extending a Requirement Formalization Metamodel .....	39
3.4.3. Two Versions of the Requirement Formalization Metamodel.....	39
3.4.4. An Example of a Formalized Requirement.....	41
3.4.5. Requirement Prioritization .....	43
<b>3.5. Summary of the Section.....</b>	<b>46</b>
<b>4. USER-DESCRIBING PROFILES IN OLAP .....</b>	<b>47</b>
<b>4.1. The Intent of the Section .....</b>	<b>47</b>
<b>4.2. The Concept of User-describing Profiles .....</b>	<b>47</b>
<b>4.3. The Method for Construction of User-describing Profiles .....</b>	<b>48</b>
4.3.1. User-describing Profile Connections and Data Sources .....	51
4.3.2. A Concept of the Preferential Profile.....	52
4.3.3. A Concept of the Recommendation Profile .....	54
<b>4.4. Summary of the Section.....</b>	<b>55</b>
<b>5. OLAP REPORTING TOOL AND ITS METADATA.....</b>	<b>57</b>
<b>5.1. The Intent of the Section .....</b>	<b>57</b>
<b>5.2. Metadata Layers .....</b>	<b>57</b>
5.2.1. Physical Metadata .....	58
5.2.2. Logical Metadata.....	59
5.2.3. Reporting Metadata.....	60
5.2.4. Semantic Metadata.....	61
5.2.5. OLAP Preferences Metadata.....	62
<b>5.3. Technical Details on the OLAP Reporting Tool .....</b>	<b>68</b>
<b>5.4. Summary of the Section.....</b>	<b>69</b>
<b>6. METHODS FOR GENERATION OF RECOMMENDATIONS IN THE OLAP REPORTING TOOL .....</b>	<b>70</b>

<b>6.1. The Intent of the Section .....</b>	<b>70</b>
<b>6.2. The Proposed Methods for Providing Report Recommendations .....</b>	<b>70</b>
6.2.1. Hot-Start Method .....	71
6.2.2. Cold-Start Method.....	78
6.2.3. Semantic Hot-Start Method.....	81
6.2.4. Adding a Recommendation Component .....	90
6.2.5. Examples of Generated Recommendations .....	91
<b>6.3. Summary of the Section.....</b>	<b>97</b>
<b>7. AN EMPIRICAL STUDY TO EVALUATE METHODS FOR GENERATION OF RECOMMENDATIONS .....</b>	<b>99</b>
<b>7.1. The Intent of the Section .....</b>	<b>99</b>
<b>7.2. The Goal of the Experimentation and Research Questions .....</b>	<b>99</b>
7.2.1. The Goal of the Experimentation.....	99
7.2.2. Research Questions .....	99
7.2.3. Phylosophical Stance .....	101
<b>7.3. Research Methodology .....</b>	<b>101</b>
7.3.1. Context of the Experimental Study .....	101
7.3.2. Subjects .....	102
7.3.3. Variables .....	103
7.3.4. Design Principles .....	106
7.3.5. Conducting the Experiment and Data Collection.....	108
<b>7.4. Experimentation Results .....</b>	<b>109</b>
7.4.1. Results of the Log-table Analysis .....	109
7.4.2. Results of the User Survey Represented Graphically .....	114
7.4.3. Reporting Results of the User Feedback .....	124
<b>7.5. Summary of the Section.....</b>	<b>128</b>
<b>8. CONCLUSIONS.....</b>	<b>130</b>
<b>8.1. Results of the Research.....</b>	<b>130</b>
<b>8.2. Conclusions on the Experimental Study .....</b>	<b>132</b>
<b>8.3. Conclusions on the Research Goal and Formulated Hypotheses .....</b>	<b>133</b>
<b>8.4. Discussions and Limitations on the Research .....</b>	<b>134</b>
<b>ACKNOWLEDGMENTS.....</b>	<b>136</b>
<b>REFERENCES .....</b>	<b>137</b>
<b>APPENDICES.....</b>	<b>146</b>
<b>Appendix 1. Experimentation tasks for student user group.....</b>	<b>146</b>
<b>Appendix 2. Experimentation tasks for academic staff user group.....</b>	<b>148</b>
<b>Appendix 3. Experimentation tasks for administrative staff user group.....</b>	<b>150</b>
<b>Appendix 4. User guide for report execution in different recommendation modes.....</b>	<b>152</b>
<b>Appendix 5. User survey on report execution in different recommendation modes.....</b>	<b>155</b>
<b>Appendix 6. User survey results grouped by user experience.....</b>	<b>157</b>

## 1. INTRODUCTION

In the course of time Touch-screen cellphones, laptops and other devices have become indispensable and widely used in daily life. The overall amount of data is dramatically increasing from year to year, thus, leading to overload with data. For instance, according to mobile data traffic forecast<sup>1</sup> by Cisco, 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" [Inm02]. 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.

During working sessions with OLAP applications the working patterns can vary. Due to the large volumes of data the typical OLAP queries performed via OLAP operations by users may return too much information that sometimes makes data exploration a tedious and time-consuming task. If there are too many constraints, the result set can be empty. In other cases, when the user explores previously unknown data, OLAP query result may differ from user's expectations. Moreover, a user is rather limited in expressing his/her likes and dislikes to get the results that are more satisfying. However, there is a space for experiments in personalization opportunities in OLAP with the purpose to provide user with data that is relevant for him/her.

In business dictionary [BD] *personalization* is defined as "creation of custom-tailored services that meet the individual customer's particular needs or preferences". Personalization can be provided by adjusting data and its visualization according to user preferences. In terms of this thesis *user preferences* are constraints of a certain type (see section 2.2.4.), which are applicable to OLAP schema, report data, and report visual layout. Each user preference is assigned a number to indicate the importance of the given constraint.

Marcel [Mar12] 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  $q' \subset q$  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

---

<sup>1</sup> Cisco mobile data traffic forecast available at:  
[http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html)

*recommendation* is: “given a database query  $q$  and some user profile compute a query  $q'$  such that neither  $q' \subset q$  nor  $q \subset q'$  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`.

Introduction of personalization into a system may be achieved in two ways:

- A user may manually alter his/her preferences, so that visual layout and data would be up to the user needs;
- A user does not have to alter his/her preferences, because user preferences are defined by the system itself and its visual layout and data is adapted accordingly by the system. In [Wei03] it is said that in a personalized system user data may be acquired taking into consideration user activity.

In [Kim02] the author gives at least two reasons, why personalization is worth drawing attention to. First, there is a large amount of data accessible for a user, which is why it is essential to deliver the data that is relevant to a particular user or a group of users. This data has to be selected and sorted depending on user needs. Second, personalization introduction in business context ties up marketing and individual customer groups. Thus, income increases, when a customer receives relevant and timely recommendations on certain goods or services.

When speaking about personalization in OLAP, one takes for granted that there exists some data warehouse to collect multidimensional data, however, usually no particular attention is paid to the quality of the conceptual model. It is worth to notice that to accumulate the data of interest, the conceptual model of a data warehouse should comply with user requirements. This thesis also addresses such questions as, for example, how to capture, structure, and process user requirements to leverage the development of the conceptual model of a data warehouse.

## 1.1. Motivation, Topicality and Novelty of the Subject

### 1.1.1. Motivation

Although there exist studies initiated by [Kie02, Cho02] and continued by other researchers on user preferences in the field of databases, personalization in data warehouses still deserves more attention by researchers (as stated in [GR09a]) and remains a field to be explored more thoroughly both on theoretical and practical level.

As mentioned in [GR09a], there are various reasons for making a research on the subject of OLAP personalization. First of all, user preferences allow a user to focus on the data that seems to be the most essential. Typically, data warehouses store large amounts of data which increases over time. While composing and executing queries, user preferences would be a natural way how to avoid both an empty set of results and data flooding. Data evaluation and ranking in accordance with user preferences would allow to solve both of the above-mentioned issues. Secondly, during OLAP sessions a user might not know exactly what kind of data he/she is looking for, thus, preferences allow user to specify a pattern of what data to select. As user preferences are expressed by *soft constraints* (see section 2.2.4.), even in case when there is no data that strictly matches the pattern some data is returned and is ranked by its relevancy to user preferences. Thirdly, it would be worthwhile to give a user an opportunity to express preferences on aggregated data. Data warehouse serves for providing users with aggregated data, grouping it at different hierarchy levels. The level of aggregation is of high importance, because it has an impact on result data that may turn out to be of not much use for being either too detailed or too general. For that reason, [GR09a] claim that users should be given an opportunity to express their preferences to data grouped on a particular hierarchy level, for instance, indicating that data aggregated by months is preferred to daily or yearly aggregated data.

The experience in using standard commercial applications for producing and managing data warehouse reports (for instance, Oracle Business Intelligence Discoverer<sup>2</sup> and MicroStrategy<sup>3</sup>) at the University of Latvia as well as participation in scientific projects and development of a new data warehouse (or OLAP) reporting tool [Sol07] served as a complimentary motivation for further studies in the field of OLAP personalization. The new OLAP reporting tool is a suitable environment for implementing and testing the developed techniques of OLAP personalization. In this case, recommendations on OLAP reports are implemented so that the users of the reporting tool not only would create, modify, and execute

---

<sup>2</sup> Oracle Business Intelligence Discoverer available at: <http://www.oracle.com/technetwork/developer-tools/discoverer/overview/index.html>

<sup>3</sup> MicroStrategy available at: <http://www.microstrategy.com/software/products/report-services>

reports on data warehouse schema, but also get some guidance on what else to examine. 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.

### **1.1.2. *Topicality and Novelty***

The field of personalization in OLAP still is being explored among the researchers worldwide. The papers on this topic are discussed at such international conferences as ICEIS, ADBIS, DaWaK, BIR, ACM TODS, ACM SIGIR, ACM SIGMOD, ACM SAC, ACM RecSys, CAiSE, and ICDE as well as published in such scientific journals as IJCSI, IJESI, IJDMS, and IEEE and LNCS proceedings. The ACM 16th international workshop on Data Warehousing and OLAP (DOLAP'13) had personalization in data warehouses as one of its topics (held in Burlingame, CA, USA).

One of the recent comparative studies of OLAP personalization approaches was conducted by [AG12]. The authors analyzed data warehouse personalization techniques according to such criteria as user characteristics, user context, user behavior, user requirements, and user preferences. In that study the authors pointed out three possible fields for the further research: (i) user preferences are more often extracted explicitly rather than implicitly, which might be quite disturbing for a user, thus, more attention should be paid to implicit profiling techniques; (ii) user requirements are, in fact, a personalization factor, which is not fully exploited in data warehouses, which is why the authors advise to take into account user requirements; and (iii) currently there is no approach that would alone provide a multi-faceted personalization, i.e. on the level of schema, interaction, and visualization. This thesis addresses the 1st and the 2nd concern highlighted by [AG12] as well as the 3rd one in the aspect of interaction.

Also, in 2012 a lecture “OLAP Query Personalisation and Recommendation: An Introduction” by Marcel [Mar12] was published with an aim to describe how personalization and recommendation techniques can be applied in OLAP context. Later, in 2014, Marcel [Mar14] presented a paper on query log exploration to examine user preferences, navigational habits, and discoveries made during former sessions.

In [KB13] authors propose to create a data warehouse materialized view for each user with respect to his/her profile. One of the latest papers on the subject of personalization in OLAP was presented at a large international ICEIS conference in 2013 and was dedicated to adapting dimension hierarchies by clustering given dimension hierarchy instances according

to user needs [BK13].

Some researchers in their latest works – [AG12, BK13, KB13] – refer to the paper [KN10], co-author of which is the author of this thesis.

All of the above-mentioned facts confirm the topicality of the subject of personalization in data warehouses.

The author of this thesis considers that scientific novelty of the research presented in the thesis is as follows:

- The ability to express preferences on the level of OLAP schema elements would be beneficial for a user who is unfamiliar with the structure of data warehouse report or uncertain about the data of interest, however, as the results of the literature studies have shown, neither of OLAP query recommendation techniques generates recommendations analyzing OLAP schema and its elements (or logical metadata; see definitions in section 2.2.1.). The methods developed by the author produce report recommendations taking as an input OLAP schema elements and are suitable for different groups of users – novice, advanced or expert;
- A metamodel to describe user preferences is compatible with logical, physical, and semantic metadata of the data warehouse based on CWM (*Common Warehouse Metamodel*, [CWM]) standard, which means that preference metadata can be integrated in some other reporting tool that supports multidimensional structure of the data to take advantage of user preferences;
- As mentioned in [AG12], user requirements are indeed a personalization factor, currently not extensively employed in OLAP field. A metamodel to formalize data warehouse information requirements that affects the construction of the conceptual model of a data warehouse was approbated by means of a case study and presented in this thesis.

## 1.2. Goals and Tasks of the Thesis

The goal of this doctoral thesis is to provide new methods to support personalization in the OLAP reporting tool delivering data that satisfies user needs.

In order to reach this goal the following tasks should be accomplished:

1. To perform a literature study of the state-of-the-art directions in data warehouse personalization and develop a way to classify and compare them with an aim to identify a gap in research and determine the direction for introducing personalization into the experimental environment (i.e. the new OLAP reporting tool);



2. To consider preliminaries before introducing OLAP personalization targeted at construction of the conceptual model of a data warehouse that would satisfy user needs;
3. To bring forward a model that describes a user of a data warehouse with a set of generic profiles (e.g. temporal, spatial, interaction) and covers various aspects of OLAP personalization;
4. To develop a metamodel for OLAP user preferences in the reporting tool and to integrate OLAP preferences into the reporting tool so that OLAP preferences metadata layer would be compatible with the remaining layers of metadata of the reporting tool (i.e. logical, physical, report, and semantic);
5. To present new methods for data warehouse personalization and implement them in the experimental environment (i.e. OLAP reporting tool);
6. To prepare the reporting tool for a set of experiments (for instance, to load data into the data warehouse and create reports);
7. To develop a thorough plan of the experimentation and describe context, subjects, variables, design principles, execution and data collection, and data analysis;
8. To test methods proposed in terms of this thesis by executing an experiment in laboratory settings with a set of subjects belonging to different groups of users (students, academic staff, and administrative staff);
9. To gather and evaluate results of experimentation with respect to performance of each of the methods from the point of view of the researcher in the context of laboratory settings.

### **1.3. Hypotheses Formulated in the Research**

In terms of the thesis the following hypotheses were suggested:

- Integration of personalization into the data warehouse reporting tool can save effort of the user during the working sessions with the reporting tool;
- Methods for generation of recommendations in OLAP that take as input user preferences gathered implicitly or explicitly and are suitable for different groups of users may be proposed.

## 1.4. Research Methods Applied

Both theoretical and empirical methods were applied in this thesis:

- A literature review was performed to study directions of data warehouse personalization followed by a comparative analysis of the approaches according to certain criteria;
- Zachman Framework [Zac, Zac03] was applied to develop a set of generic user-describing profiles (user, interaction, temporal, spatial, preferential, and recommendational) as well as to construct sets of attributes of user-describing profiles. To construct the above-mentioned profiles, literature studies of such sources of information as data warehouse literature, CWM standard [CWM], scientific and technical papers, along with empirical studies of the data warehouse of the University of Latvia, Oracle Warehouse Builder, and others were performed;
- Modeling methods were applied to develop OLAP preferences metamodel, user-describing profiles, and requirement formalization metamodel;
- A data mart to gather data for the experimentation has been designed and implemented and is currently being maintained and updated with real data on study process in the University of Latvia;
- A recommendation component that includes three methods for generation of report recommendations was implemented in the reporting tool;
- An empirical study (which was planned consulting the guidelines for conducting an experimental study [Bas92, KPP02, WHH03, ESSD08]) was performed to analyze and evaluate methods for generation of report recommendations in the reporting tool with precision/recall technique and statistical tools.

## 1.5. Main Results of the Research

The main results of this doctoral thesis are the following:

- Four approaches for introducing personalization in OLAP were highlighted: preference constructors (*PC*), rule-based personalization (*RBP*), visual OLAP (*VO*), and recommendations (*R*). A comparative analysis was performed in order to point out (i) the level of personalization as well as personalization options described and its applicability to OLAP schema elements, aggregate functions, and OLAP operations, (ii) the type of constraints (hard, soft or other) used in each approach, (iii) the methods for obtaining user preferences and collecting user information. A gap and a subject for a new study was defined as generating recommendations in a data warehouse

reporting tool having logical metadata as an input, unlike in other recommendation-based OLAP personalization approaches.

- Apart from OLAP personalization opportunities, such aspect as the delivery of the data of interest to a user by means of constructing the conceptual model of a data warehouse that satisfies user requirements was considered. Special attention was paid to the development of the formal requirement repository, and an extended metamodel to formalize information requirements was presented.
- A method has been proposed, which provides an exhaustive description of interaction between a user and a data warehouse using the concept of Zachman Framework [Zac, Zac03]. In accordance with this framework a composite user profile consisting of a set of generic user-describing profiles (user, interaction, temporal, spatial, preferential and recommendational) has been developed.
- A metamodel to formulate user preferences for OLAP schema elements and aggregate functions – OLAP preferences metamodel – has been proposed based on the empirical studies of reporting tools. OLAP preferences metadata got integrated with other metadata layers of the OLAP reporting tool [Sol08a], i.e. logical, physical, report, and semantic.
- Three distinct content-based methods for construction of report recommendations have been developed: *hot-start* method that takes advantage of the user activity log, *cold-start* method that defines similarity of reports based on their structure, and *semantic hot-start* method that employs user-defined preferences for report elements. Recommendations are generated based on preference information in user profile, which is updated either implicitly or explicitly depending on the method. A recommendation component that includes implementation of these methods has been added to the reporting tool.
- The experimental study was performed in laboratory settings involving 30 subjects with various level of experience with reporting tools (novice/advanced user/expert) with an aim to explore which of the methods for generating recommendations in the reporting tool would produce more accurate recommendations. A data mart to gather data on user interaction with Moodle course management system (referred as Moodle or Moodle CMS) and study process in the University of Latvia was designed and developed as well as 70 reports were created for an experimental study. 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.

## 1.6. Approbation of the Results

The theoretical part of the thesis was developed in terms of the project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044 supported by the European Social Fund. The recommendation component that includes methods to produce report recommendations (i.e. hot-start, cold-start, and semantic hot-start) taking advantage of the user preference data for OLAP schema elements and aggregate functions has been developed and integrated into the new OLAP reporting tool in terms of the project "Support for Doctoral Studies at University of Latvia" provided by the European Social Fund. An experimental study was successfully conducted to approbate the recommendation component and underlying methods in a testing environment (i.e. the new OLAP reporting tool) involving 30 subjects with different rights and experience with a maximum number of 70 reports available.

The results of the study described in this doctoral thesis are published in the following 8 papers, to the creation of which the author had contributed significantly:

An overview that covers different directions of OLAP personalization, its characteristics and a comparative analysis is reflected in:

1. [KN11] Kozmina, N., Niedrite, L. 'Research Directions of OLAP Personalization'. In Proceedings of the 19th International Conference on Information Systems Development (ISD'10), Prague, Czech Republic. Springer Science+Business Media, 2011, pp. 345-356. (indexed in Scopus)

A metamodel and its extended version to formalize information requirements with an aim to employ them for the development of the conceptual model of a data warehouse and to raise the quality of the subsequent OLAP personalization were presented in:

2. [NNK11] Niedritis, A., Niedrite, L., Kozmina, N. 'Performance Measurement Framework with Formal Indicator Definitions'. In: J. Grabis, M. Kirikova (eds.) Perspectives in Business Informatics Research, LNBIP, vol. 90, Springer, Berlin, 2011, pp. 44-58. (indexed in Scopus and ISI)
3. [KN14] Kozmina, N., Niedrite, L. 'Extending a Metamodel for Formalization of Data Warehouse Requirements'. In: B. Johansson et al. (eds.) Perspectives in Business Informatics Research, LNBIP, vol. 194, Springer, Berlin, 2014, pp. 362-374.

A set of generic data warehouse user-describing profiles was proposed in:

4. [KN10] Kozmina, N., Niedrite, L. 'OLAP Personalization with User-Describing Profiles'. In Proceedings of the 9th International Conference on Perspectives in

Business Informatics Research (BIR'10), Rostock, Germany. Springer, Heidelberg, 2010, LNBIP, vol. 64, pp. 188-202. (indexed in Scopus and ISI)

User preference integration with other metadata of the reporting tool, an approach to determine user preferences from semantic metadata, and new methods (hot-start and cold-start) to define user preferences implicitly are described respectively in:

5. [KS12] Kozmina, N., Solodovnikova, D. 'Towards Introducing User Preferences in OLAP Reporting Tool'. In: Niedrite L, et al. (eds.) BIR 2011 Workshops, Riga, Latvia. Springer, Heidelberg, 2012, LNBIP, vol. 106, pp. 209-222. (indexed in Scopus and ISI)
6. [SK11] Solodovnikova, D., Kozmina, N. '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, 2011, pp. 363-370.
7. [KS11] Kozmina, N., Solodovnikova, D. '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, 2011, 46:35-42. (indexed in EBSCO, ProQuest and VINITI databases)

Implementation of the recommendation component that produces recommendations in the reporting tool by means of hot-start and cold-start methods is reflected in:

8. [Koz13] Kozmina, N. 'Adding Recommendations to OLAP Reporting Tool'. In Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS'13), Angers, France, 2013, vol. 1, pp. 238-245. (indexed in Scopus and ISI)

The results of this thesis were presented by the author of this thesis at international scientific conferences:

- BIR (Perspectives in Business Informatics Research) in 2010 [KN10], 2011 [SK11], and 2014 [KN14];
- ISD (Information Systems Development) in 2010 [KN11];
- ICEIS (International Conference on Enterprise Information Systems) in 2013 [Koz13];

The results of this thesis were presented by the author of this thesis at scientific conferences of local level:

- The 68th Scientific Conference of the University of Latvia in 2010, Information Technology section, presentation “OLAP personalizācijas pētījumu virzieni” (“Research Directions in OLAP Personalization”);
- The 69th Scientific Conference of the University of Latvia in 2011, Information Technology section, presentation “Datu noliktavu pētījumi Latvijas Universitātē” (“Data Warehouse Research Study in the University of Latvia”);
- A conference in terms of ESF project Nr.2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044 in 2011, presentation “Personalizācijas un evolūcijas atbalsts datu noliktavas atskaišu rīkā” (“Personalization and Evolution Support in Data Warehouse Reporting Tool”);
- A final conference in terms of ESF project Nr.2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044 in 2012, presentation “Procesu datu noliktavu pētījumi: modeļi, personalizācija, evolūcija” (“A Research in Process Data Warehouses: Models, Personalization, Evolution”);
- The 71st Scientific Conference of the University of Latvia in 2013, Information Technology section, presentation “Personalizācijas iespējas datu noliktavās” (“Personalization Opportunities in Data Warehouses”).

The author of this thesis participated as a co-author and was represented at the international scientific conference BIR (Perspectives in Business Informatics Research) in 2011 [KS11, NNK11, KS12].

## **1.7. Outline of the Thesis**

This thesis consists of 158 pages containing 33 figures, 22 tables, references, and 6 appendices. The thesis is composed of 8 sections including introduction and conclusions. The rest of the thesis is organized the following way.

Section 2 summarizes the literature review that was performed in the field of data warehouse (OLAP) personalization, highlighting four existing research directions. An evaluation has been done to point out personalization options provided and its applicability to OLAP schema, the type of constraints used in each approach, and methods for obtaining user preferences and collecting user information. The goal of the literature review was to classify the ideas already proposed in the field of OLAP personalization to find a direction that can be

followed to develop new features of OLAP personalization and implement it in the reporting tool.

In section 3 an extended metamodel to formalize information requirements is presented as a result of a case study that included over 150 information requirements for the currently operating data warehouse of the University of Latvia. The requirement formalization metamodel contributes to the development of the conceptual model of a data warehouse, the quality of which, in its turn, has an impact on further data warehouse personalization.

In section 4 a model that describes a data warehouse user with a set of generic profiles in order to cover various aspects of data warehouse user interaction with the system is set forward. The basic idea of development of user-describing profiles was inherited from Zachman Framework concept [Zac, Zac03].

Section 5 describes OLAP reporting tool developed at the University of Latvia, which is considered as an experimental environment for introducing OLAP personalization, providing technical details on the implemented OLAP reporting tool as well presenting its metadata that consists of five interconnected layers: logical, physical, reporting, semantic, and OLAP preferences metadata.

Section 6 presents content-based methods (cold-start, hot-start, and semantic hot-start) and its underlying algorithms for construction of recommendations for reports developed by the author of this thesis. Taking advantage of data on user preferences for data warehouse schema elements, existing reports that potentially may be interesting to the user are distinguished and recommended. Cold-start, hot-start, and semantic hot-start methods are implemented in the recommendation component of the OLAP reporting tool.

In section 7 a detailed plan and the results of the experimentation in the OLAP reporting tool are given. The experimental study targeted to explore which of the methods for generating recommendations in the reporting tool has a deeper impact on users (i.e. produces more accurate recommendations) was performed in laboratory settings.

Section 8 is the concluding one, where the results of this thesis are summarized.

## **2. LITERATURE REVIEW ON DATA WAREHOUSE PERSONALIZATION**

### **2.1. The Intent of the Section**

The intent of this section is to summarize the literature review that was performed in the field of data warehouse (OLAP) personalization. Though the initial version of this literature review was published in [KN11], nevertheless, it was revised and supplemented with more up-to-date papers. Four existing research directions for introducing personalization in OLAP have been highlighted: preference constructors, rule-based personalization, visual OLAP, and recommendations. The goal of the literature review and succeeding comparative analysis was to classify and characterize the approaches already proposed in the field of OLAP personalization to find out an unexplored problem or a gap in research, and decide on a direction that can be followed to develop new methods of OLAP personalization.

### **2.2. Research Directions in OLAP Personalization**

Different OLAP personalization types – OLAP query personalization, personalization during runtime, visual personalization of query results, etc. – are described in this section. A summary of various approaches related to each direction of OLAP personalization is proposed in terms of this section followed by a comparative analysis of these approaches according to some criteria. An evaluation has been provided in order to point out (i) the level of personalization as well as personalization options described and its applicability to OLAP schema elements, aggregate functions, and OLAP operations, (ii) the type of constraints (hard, soft or other) used in each approach, (iii) the methods for obtaining user preferences and collecting user information.

#### **2.2.1. OLAP Schema, its Elements and Basic OLAP Operations**

Let's define the OLAP schema and its elements – dimensions and its attributes, hierarchies and its levels, fact tables and its measures. In terms of this thesis, OLAP schema and its elements are also referred as *logical metadata*.

An OLAP schema (also multidimensional or data warehouse schema) is employed to model a data warehouse. An *OLAP schema* is a collection of database objects, including tables, views, indexes, and synonyms [LSS05]. The simplest data warehouse schema is the star schema, which is called so, because its graphical representation resembles a star. The center of the star consists of a fact table and the points of the star are the dimension tables. However, there are other OLAP schema models also exploited in data warehouses, e.g. the



snowflake schema (a star schema with normalized dimensions), the constellation schema (a combination of several star schemas, which occurs when the number of dimensions are shared). A measure is the main object of the OLAP schema.

Instances of a *fact* correspond to events that occurred [GR09b]. For example, every single sale is an event. A *measure* is a numerical property of a fact, and describes one of its quantitative aspects of interests for analysis [WK07]. For instance, each purchase may be measured by the number of units sold. Typically, measures are numerical, because they are used for computations.

A *fact table* contains either detail-level facts or facts that have been aggregated [LSS05]. Typically, a fact table has two types of columns: the ones that contain numeric facts (measurements), and the ones that are foreign keys to dimension tables. A fact table usually contains facts with the same level of aggregation. To aggregate data, aggregate functions are applied (e.g. SUM, COUNT, AVG).

A *dimension* is a fact property with a finite domain and describes one of its analysis coordinates [WK07]. It is also defined as a structure, often composed of one or more hierarchies, that categorizes data. Dimension data is typically collected at the lowest level of detail and then aggregated into higher-level totals often used for analysis.

A dimension *attribute* is a property with a finite domain of a dimension [WK07]. Dimension attributes help to describe the dimensional value, and usually contain descriptive and textual information.

*Hierarchies* are logical structures that use ordered levels as a means of organizing data, and a *level* represents a position in a hierarchy [LSS05]. A hierarchy can be used to define data aggregation: for example, in a time dimension a hierarchy might aggregate data from the month level to the quarter level to the year level. Each level is logically connected to the levels above and below it within a hierarchy. Level relationships specify top-to-bottom ordering of levels from most general to most specific information.

*Drill-down* and *roll-up* are the operations for moving down and up along the dimensional hierarchy levels [BHS+98]. With drill-down users can navigate to higher levels of detail, while with roll-up they can zoom out to see a summarized level of data. The hierarchies within dimensions determine the navigation path. These are the basic OLAP operations.

## 2.2.2. A Description of OLAP Personalization Directions

### Preference Constructors

OLAP query personalization with Preference Constructors (PC) reminds of an approach to define user preferences in database queries proposed and developed by [Kie02]. Algebra that allows formulation of preferences on attributes, measures, and hierarchies is defined in [GR09a]. An important feature of the proposed algebra is an opportunity to express preferences for hierarchy levels of group-by sets, which consequently leads to expressing preferences for facts. A roll-up function is used to outspread preferences applied to attributes along the whole hierarchy. Preferences can be defined on both attributes and measures, i.e. on categorical or numerical attributes.

Consider two types of preferences: *base* and *complex* [GR09a, KEW11, ERHK14]. In base preferences constructors are applied to attribute, measure, and hierarchy level. Complex preferences consist of the combination of base preferences, which can be expressed by means of the formal grammar. Base preference constructor in this grammar is one of predefined operators like POS, NEG, BETWEEN or others. One may describe both types of preferences by means of formal grammar as follows:

$$\langle \text{expr} \rangle := \langle \text{baseConstr} \rangle \mid \langle \text{expr} \rangle \otimes \langle \text{baseConstr} \rangle$$

$$\langle \text{baseConstr} \rangle := \text{POS} \mid \text{NEG} \mid \text{BETWEEN} \mid \text{LOWEST} \mid \text{HIGHEST} \mid \text{CONTAIN} \mid \text{NEAR} \mid \text{COARSEST} \mid \text{FINEST} \mid \text{NEARBY} \mid \text{AROUND} \mid \text{WITHIN} \mid \text{MORE THAN} \mid \text{LESS THAN} \mid \text{ION ROUTE} \mid \text{LAYERED} \mid \text{EXPLICIT} \mid,$$

where  $\langle \text{baseConstr} \rangle$  stands for base preference,  $\langle \text{expr} \rangle$  stands for complex preference, and  $\otimes$  denotes Pareto operator, which is used to combine multiple base preferences, thus, completing a complex preference. Pareto composition or  $P1 \otimes P2$  signifies that preferences  $P1$  and  $P2$  are perceived as equally important, and it is commutative and associative.

Let's consider several base preference examples. For instance, a base preference to indicate interest of a user for some attribute is expressed using a preference constructor  $\text{POS}(\text{Month}, \text{'Sep-14'})$ , where  $\text{Month}$  is an attribute itself and  $\text{'Sep-14'}$  is a value of the given attribute. The peculiarity of setting base preferences to attributes is the following: if an attribute in a given preference is also a level of some hierarchy, then the preference is propagated to all levels of the corresponding hierarchy. In terms of this example, the Time hierarchy includes such levels as  $\text{Day} \rightarrow \text{Month} \rightarrow \text{Year}$ , where  $\rightarrow$  is a roll-up function over this hierarchy. This way,  $\text{POS}(\text{Month}, \text{'Sep-14'})$  means that all facts aggregated on the level of the month September 2014, as well as all facts that refer to each day of September 2014, and

all facts aggregated on the level of the year 2014 are preferred to all other facts. Another example is CONTAIN(Time, Year) that expresses a preference for facts aggregated on the Year hierarchy level of Time hierarchy.

Preference constructors are implemented as Preference SQL [KEW11, ERHK14] that consists of Standard SQL structures and preferences [Kie06]. Preference queries are specified in [HK05] using a SELECT-FROM-WHERE part to state conditions in WHERE-clause and a PREFERRING-GROUPING part to express preferences in a query. In both parts of a preference query AND can be used to combine more than one constraint, but in the PREFERRING-clause it has a meaning of Pareto operator. In this case AND prescribes combination of equally important preferences.

An example of preferences stated in Preference SQL is adapted from [ERHK14] and explained below. Assume that a user would like to express a wish for a car having the *highest power* and a *price around 35000 EUR* for each group of the registration year:

```
SELECT id, power, price, year FROM car
PREFERRING power HIGHEST AND price AROUND 35000
GROUPING year ORDER BY year.
```

Here the PREFERRING-clause includes a Pareto composition of two preferences (i.e. both of the preferences are equally important to the user). Preference constructors for each of the preferences are: HIGHEST(power) and AROUND(price, 35000). The difference between GROUPING operator in Preference SQL and GROUP BY operator in standard SQL is better demonstrated with the example: the results of GROUPING return cars grouped by year, whereas within each of the groups the results that comply, first, with both, then, with one of the preferences are shown prior to others; meanwhile, GROUP BY would solely group the results by year.

### ***Rule-based Personalization***

Rule-based personalization is subdivided into two groups: dynamic rule-based personalization (*RBP-D*) and rule-based personalization with constraints (*RBP-C*).

Let's consider dynamic rule-based personalization (*RBP-D*) first. The time and method of creation of an adapted OLAP fact table define the type of personalization – static or dynamic. Static OLAP personalization means that for different users of the data warehouse diverse OLAP fact tables are created during design time. Dynamic OLAP personalization means that an adapted OLAP fact table is created during the user session time according to the needs and performed actions of the user. In [GPMT09] the authors cover dynamic OLAP

personalization, because it is a more complicated task as it involves explicit or implicit interaction with the user. To specify OLAP personalization rules, authors suggest employing PRML or Personalization Rule Modeling Language, which is a method-independent personalization specification language described in [GG06]. PRML is actually based on Event-Condition-Action-rules or ECA-rules, which are described in [TSM01]. The structure of such PRML rule can be presented with following statement:

WHEN *event* DO IF *condition* THEN *action* ENDIF ENDWHEN.

The knowledge about each user is captured in the user model and includes such information as, for instance, user characteristics such as language, role or department, user context such as location, time or browsing device, user browsing behavior, etc. User model is being supplemented with information gathered at runtime by means of PRML rules. There are two kinds of actions to be used in personalization rules in [GPMT09]. In order to get information about the user during runtime and update the user model or to update values of dimension attributes and fact table measures, a *set*-action is used, e.g. for calculating user's interest in certain hierarchy levels (which in [GPMT09] is measured as the number of times the user moved from the finer level of granularity to a coarser one applying the roll-up function). To personalize multidimensional model, *hide*-actions are used on OLAP schema objects, e.g. a hide-action may be executed, if the user's degree of interest in a certain hierarchy level is lower than a pre-defined value.

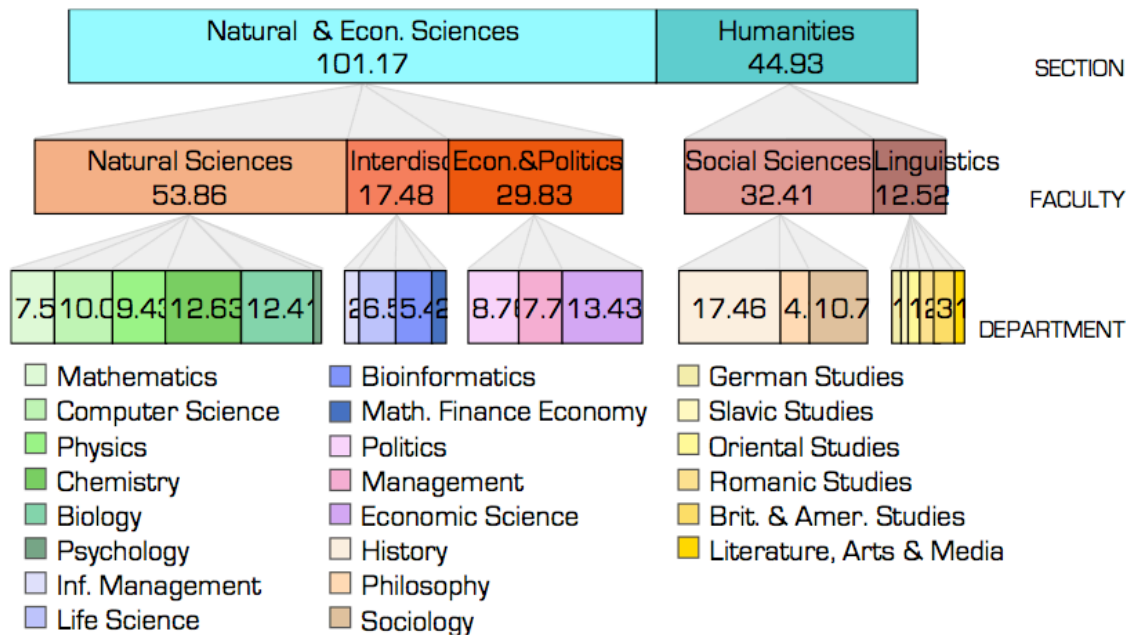
Next, let's turn to rule-based personalization with constraints (RBP-C) recently described in [BK13]. In terms of this method existing hierarchies are supplemented with new hierarchy levels based on current user preferences, which are represented by constraints that a user defines. Hierarchy level data is grouped in clusters with the help of PRoCK operator (*Personalized Roll-up Operator with Constrained K-means*), which employs some clustering algorithm (i.e. K-means clustering method) that comes from data mining. Then, users express their preferences about the obtained clusters with *must-link* and *cannot-link* constraints, thus, forming a new granularity level in the considered hierarchy. A *must-link* constraint indicates that two data instances must be grouped in the same cluster, whereas a *cannot-link* constraint denotes that two instances must not be grouped in the same cluster. This way, a user may obtain a personalized hierarchy (a new level of which will be added to a corresponding dimension) to target his/her analysis needs.

An example in [BK13] that illustrates the rule-based personalization with constraints goes as follows. Assume that there is a hierarchy Country  $\rightarrow$  Continent, where  $\rightarrow$  is a roll-up function over this hierarchy. One may wish to measure the impact of the Internet for each of 9 African countries by measuring the number of Internet users in relation to the population the

country. These parameters are passed to the algorithm that creates clusters of data with PRoCK operator and adds a new level to the hierarchy: Country → CountryGroup. If a country due to some reason should be regrouped and placed into another cluster, then a user may express these preferences by stating must-link or cannot-link constraints.

**Visual OLAP**

Visual personalization of OLAP fact table – Visual OLAP (VO) – may also be considered as a means of personalization. The concept of Visual OLAP is disburdening the user from composing queries in “raw” database syntax (SQL, MDX), whereas events like clicking and dragging are transformed into valid queries and executed [MS08]. In terms of Visual OLAP approach, visualization is perceived as the key method for both query specification and data exploratory analysis.



**Fig. 2.2.2.1.** A decomposition tree example demonstrating aggregated measures on three levels of granularity – Section, Faculty, and Department [MSKM07]

In [MSKM07, JRTZ09, RT09] authors present a user interface for OLAP, where user is explicitly involved. In [MSKM07] users are able to navigate in dimensional hierarchies using a schema-based data browser, whereas in [JRTZ09, RT09] users are provided with an interface for formulating queries by means of manipulation with graphical OLAP schema and rules. A user composes a query when he/she selects a measure and an aggregation function [MSKM07]. Dimensions for “drilling down” are chosen and the values are set as filters. Having selected a measure and an aggregate function, the user simply drags any dimension

folder into the visualization area to create a new level in the decomposition tree. The decomposition tree is gained from an aggregated measure as a root splitting it along chosen dimensions. An example of one of the layouts of the decomposition trees is seen in Figure 2.2.2.1; other layouts are available in [MSKM07].

### **Recommendations**

Personalization by means of recommendations can be subdivided into two groups: recommendations with user session analysis (*R-SA*) and recommendations with user profile analysis (*R-PA*).

The idea of recommendations with user session analysis (*R-SA*) is described in [GMNS09, GMNS11, Mar14], 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 stated in [Mar14] is to identify the goal of user's analysis session. This can be achieved by exploring the difference between measure values in executed queries. Measure values are being compared and a significant unexpected difference in the data is being detected. The emphasis is not on recommending queries from sessions that are prior to the current session, but on recommending queries from all sessions, where a user had found the same unexpected data as in current session. In [GMNS09] the authors introduce a concept of a "drill-down (or roll-up) difference query", which is classified as such, if the result of this query confirms the difference of measure values at a lower level of detail (for drill-down) and at a higher level of detail (for roll-up).

Another recently developed approach that exploits past user experience with queries to assist in constructing new queries is presented in [KB12], and also falls into category of recommendations with user session analysis. In this case, a user can build a query being guided by the most frequently employed query elements extracted from the past queries that are connected to the current query of a user by some association rules.

An approach that generates recommendations by means of user profile analysis (*R-PA*) is presented in [JRTZ09], and later implemented as a framework for OLAP content personalization in [JRTZ11]. The authors propose a context-based method for providing users with recommendations for further exploration in [JRTZ09]. 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. An analysis context includes two disjoint sets of elements: a set of OLAP schema elements – fact tables, measures, dimensions, attributes, etc. and a set of its values. Restriction predicates, i.e. restrictions on data values of measures (associated with an aggregate function) or conditions on data values of dimension attributes, are ranked with

the relevance score (a real number in the range [0; 1]). Preferences stated in the user profile, analysis context of which matches with the analysis context of the current query, are integrated in the current query, thus, providing more customized content, and such query is recommended to the user. 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. The idea of ranking preferences is also mentioned in [RT09]. Preferences in user profiles are also employed for comparing queries and personalizing query result visualization in [BGMM06]. The approach presented in [JRTZ09] was interpreted and implemented by [CG13] to add constraints to multidimensional queries distinguishing absolute and contextual preferences and to recommend relevant queries from the log-file.

### 2.2.3. A Comparison of Existing OLAP Personalization Approaches

All previously described approaches were compared to give an overview on applying personalization of different type to OLAP schema elements, functions and typical OLAP operations. The results of the comparison are given in Table 1.

Columns of the Table 1 represent the main concepts of OLAP systems: OLAP schema elements (i.e. dimensions, attributes, hierarchies, hierarchy levels, fact tables, and measures), aggregate functions, and OLAP operations (drill-down and roll-up).

**Table 1. Applicability of personalization to OLAP objects and the level of personalization**

		<i>Pers. Object</i>								<i>Pers. Level</i>		
		<b>Dimension</b>	<b>Dimension attribute</b>	<b>Hierarchy</b>	<b>Hierarchy level</b>	<b>Fact table</b>	<b>Measure</b>	<b>Aggregate function</b>	<b>Drill-down</b>	<b>Roll-up</b>	<b>Data</b>	<b>Schema elements</b>
<i>Pers. Type</i>	<b>PC</b>	-	+	-	+	-	+	-	*	+	+	+
	<b>RBP</b>	D	+	+	+	+	+	+	+	+	+	+
		C	*	*	+	+	-	-	-	+	+	-
	<b>VO</b>	+	+	+	+	+	+	+	+	+	+	+
	<b>R</b>	SA	+	+	+	+	+	+	*	+	+	-
		PA	+	+	+	+	+	+	+	*	*	+

Rows of the Table 1 contain all previously described personalization types. The cells of the table contain a value from a set of acronyms to represent an evaluation of the personalization applicability to OLAP schema element, aggregate function or OLAP

operation: “+” stands for “applicable” as it is explicitly defined in the papers; “\*” means “derivable” as personalization applicability to OLAP schema element, aggregate function or OLAP operation can be derived taking into account other personalization aspects presented in the paper (e.g. personalization considers a roll-up operation, but drill-down operation is not mentioned in the paper, and it is implied that personalization considering drill-down is derivable, because drill-down operation is an inverse operation of roll-up); and “-” stands for “no information available”, because it is not described in the paper.

In Table 1 personalization level indicates the domain, where users express their preferences – either on a detailed level by putting restrictions on data values of attributes and measures (*data level*) or on a more general one by specifying OLAP schema elements of interest (*schema elements level*).

One may observe that personalization of OLAP schema elements is mostly present in all proposed OLAP personalization types except for preference constructors (PC) and rule-based personalization with constraints (RBP-C). In PC the way of expressing user preferences for dimensions, hierarchies, fact tables as such as well as aggregate functions is not described. Out of all OLAP schema elements in RBP-C a user may only personalize hierarchies by adding a new customized hierarchy level (and, therefore, a new dimension attribute).

All approaches without exception allow to formulate user preferences on the level of data (or content), however, only few of them (namely, PC, RBP-D, and VO) tackle the aspect of expressing interest in OLAP schema elements, i.e. on the level of logical metadata. The ability to express preferences on a more general level, i.e. on the level of OLAP schema elements (or logical metadata), would be beneficial for a user who is unfamiliar with the structure of data warehouse report or uncertain about the data of interest. Neither of OLAP query recommendation techniques generates recommendations taking logical metadata as an input. A recent survey of the existing methods for computing data warehouse query recommendations is proposed in [MN11]. Authors of the survey singled out four methods that convert a 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. However, none of the methods falling into each category involved recommendations on the analysis of OLAP schema and its elements, which made the author of this thesis choose query (or report) recommendations as the area of interest and further studies.



### 2.2.4. *Hard and Soft Constraints as User Preferences*

Although the role of the preferences was recognized in applications a long ago, the database researchers paid attention to this issue only around year 2000 [AW00, BKS01, Kie02, Cho03]. It was observed that in database queries WHERE-conditions are *hard constraints* and either the non-empty result set is returned if all the conditions are satisfied, or an empty set is returned in the opposite case. Queries with hard constraints either deliver exactly the desired object if it exists, or reject the user’s request otherwise [KK02].

The authors of [PFT03] define *soft constraints* as: “Functions that map any potential value assignment into a numerical value that indicates the preference that this value or value combination carries”. In information retrieval soft constraints are used and results are arranged according to its relevancy to initial query conditions.

The hard and soft constraints are considered as a means to express user preferences. The difference between hard and soft constraints is that soft constraints can be evaluated, whereas hard constraints can be either satisfied or not. Eventually, different approaches to use soft constraints in database queries have appeared [Kie02, Cho03], turning database queries into “preference queries”. In papers [Kie02, KK02, HK05, Kie06, KEW11, ERHK14] an implementation of the framework using the above-mentioned Preference SQL is described, which is translated to SQL, and used in several deployed applications. In Preference SQL a SELECT-FROM-WHERE part involves structures of standard SQL to state hard constraints by means of WHERE-clause, and a PREFERRING-GROUPING part involves soft constraints that express preferences in a query. The authors point out that extending SQL by preferences will enable a personalized search to gain more targeted results.

Let’s see what kind of user preferences can be expressed in each of OLAP personalization types earlier discussed.

**Table 2. OLAP Personalization types and applied constraints**

		<i>Constraint Type</i>			
		<b>Hard Constraints</b>	<b>Soft Constraints</b>	<b>Other</b>	
<i>Pers. Type</i>	<b>PC</b>	-	+	-	
	<b>RBP</b>	D	+	-	-
		C	+	-	-
	<b>VO</b>	+	-	-	
	<b>R</b>	SA	-	-	+
		PA	-	+	-

Table 2 illustrates, which method is applied in each of OLAP personalization types. A “+” or “-” sign indicates that a method is/isn’t applied in each of OLAP personalization types:

- Hard Constraints,
- Soft Constraints,
- Other (i.e. the method used cannot be categorized as Hard or Soft constraints).

Preference Constructors (PC) use soft constraints to express user’s likes and dislikes, which are implemented in the PREFERRING-clause of Preference SQL.

*Example 1 (Preference Constructors).* For instance, a user would like to obtain student activity data such as time spent on exploring course informational resources, quantity of tasks assigned and completed, grades received for completed tasks, etc. He/she is interested in a specific course named “Data Warehouses”, which is an attribute of *Course* dimension in some data warehouse. A preference constructor POS(*Course*, “Data Warehouses”) may express such preference.

*Example 2 (Preference Constructors).* Suppose there is a hierarchy *Course* → *Study Program* → *Faculty*, where → is a roll-up function over this hierarchy. *Biology Masters* is one of study programs, belonging to the *Faculty of Biology*. NEG(*StudyProgram*, “Biology Masters”) states that data that does not map to *Biology Masters* study program, does not refer to courses of *Biology Masters* study program and does not map to the *Faculty of Biology*, is preferred to all the other data.

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

*Example 3 (Dynamic Rule-based Personalization).*

**Rule:** hideBusinessTrip

WHEN SessionStart DO

IF (User.Role = “Student”) THEN

hideDescriptor(Person.BusinessTrip)

ENDIF

ENDWHEN

Also, in rule-based personalization with constraints (RBP-C), must-link and cannot-link constraints refer to hard constraints, because they unambiguously define which data elements should be placed in the same cluster and which should not.

One of the aspects of Visual OLAP (VO) is user browsing through navigational OLAP schema and filtering the OLAP schema objects to be displayed [MS08]. Users' navigation events such as clicking and dragging are translated to valid SQL-queries with WHERE-clause, which in fact is a hard constraint in standard SQL [KK02].

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

*Example 4 (Recommendations with User Session Analysis).* If there is a difference that is a drop of the sales of some kind of product from 2013 to 2014, then recommended queries will contain the same difference in values. It is assumed that neither soft nor hard constraints are used in this type of personalization. In [GMNS09] authors apply the technique that develops the ideas of DIFF operator proposed in [Sar99] and applied for explaining reasons for sudden drops or increases in data values.

In user profiles utilized for generation of recommendations (R-PA) soft constraints appear. A user may express the extent of liking or disliking, as there is a relevance score that is associated with restriction predicates on element of OLAP schema [JRTZ09]. The following example illustrates the usage of soft constraints in R-PA.

*Example 5 (Recommendations with User Profile Analysis).*

$P^{\text{Role}} = (\text{'Role} \neq \text{Guest}'; 0.9; c)$  is a preference in the user profile. In this preference 'Role  $\neq$  Guest' is a predicate, which is a condition on dimension data (in other case, a predicate may be a restriction on fact table data); 0.9 is a real number between 0 and 1 that indicates a degree of relevance (i.e. a number closer to 0 means 'less relevant', while a number closer to 1 means 'more relevant');  $c$  is an analysis context that includes analyzed measures (with aggregate functions applied) and analysis axis (dimension/attribute). Here  $c = \text{"Activity, Time/Date} \geq \text{'01/01/2014'"}'$ , which means that measures of *Activity* fact table are analyzed and *Time/Date* is an analysis axis, where *Time* is a dimension and *Date* is an attribute.  $P^{\text{Role}} = (\text{'Role} \neq \text{Guest}'; 0.9; c)$  means that user's interest to include non-guest users specified by the condition 'Role  $\neq$  Guest' into qualification of user activity in course management system is very high.

All of the given examples demonstrate the ways of setting user preferences. As it is seen from the Table 2 and from the examples, not only soft and hard constraints are employed

as a means of expressing user preferences, but also special functions (as in R-SA). Hard constraints are employed in both groups of RBP and in VO, whereas soft constraints – only in PC and R-PA. Thus, the idea of processing user preferences defined with soft constraints is suitable for further studies in the field of query (or report) recommendations with user session analysis (R-SA), and is supported in one of the methods for generation of recommendations (see section 6.2.1.) put forward in this thesis.

### **2.2.5. Approaches for Collecting User Preference Data**

Typically there are two ways of collecting information about the user – explicitly and implicitly [GSCM07]. Also, a hybrid approach is possible where explicit and implicit methods are combined.

Methodologies for *explicit* user information gathering are based on user information input about themselves and their interests. Users enter information manually or choose pre-defined values from a list. The problems arise, because the users do not like to rate the objects as they are not interested or will not receive any benefit in return. In this case an explicit user profile will be very poor. Also, [GSCM07] points out that user may not be very accurate, when providing information. User preferences may change over time, thus, making the information in the user profile outdated.

In its turn, user profiles may be built based on *implicitly* gathered information. Implicit preferences present behavioral information about the user. Analysis of server logs, search, purchase, or browsing history can generate implicit preferences. A research on acquiring user preferences implicitly is presented in [KT03]. The most attractive aspect of the implicit preferences is that data about the user can be gathered without user intervention. However, authors [GSCM07] point out some limitations, for example, the data observed by the user is not always aligned with an intention to observe it. Often the time when the data is displayed to the user is interpreted as reading time. Also, the user is unable to give negative preferences, to express negative interest or dislike, whereas mouse clicks are treated as positive interest. Sometimes during the search for essential information user clicks on irrelevant links, therefore, in many cases user interest could not be equalized to the number of clicks.

### **2.2.6. Methods for Obtaining User Preferences**

An overview of existing methods for extracting user preferences for further processing is presented in [Bur02]. However, the authors of [VPF02] supplement the list with two more methods (*questions & answers, mixed initiative*):

- *Questions & Answers (Q&A)*. Information for the user profile is collected, when user answers to the questions or fills in the form. The information in user profile stays unchanged until the user updates it.
- *Mixed initiative (MI)*. This method is also called *candidate/critique mode*. User preferences are gained as a result of proposing existing solutions to a user and receiving user evaluation. The solution is improved according to the critique and proposed to the user again until it satisfies the user. An example of a system with implemented mixed initiative approach is a system presented in [SL01], where an agent is implemented for the gathering user preferences when the user expresses his/her attitude to the observed data.
- *Content-based (CB)*. This method captures user preferences from features of objects that a user has already rated or applied. Content-based user profiles are updated, when some new user preference-related information appears.
- *Utility and Knowledge-based (UKB)*. These methods calculate similarity between what a user needs stated as preferences and what is available.
- *Collaborative (C)*. In terms of this method multiple user ratings are aggregated and compared with the rating of a particular user of a certain object.
- *Demographic (D)*. This method gathers demographic characteristics of a user. Users with similar characteristics are grouped into classes.

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

**Table 3. Methods for obtaining preferences and user information collection applied in different types of OLAP personalization**

		<i>Method for Obtaining User Preferences</i>						<i>Approach for Collecting User Preference Data</i>		
		Q&A	MI	CB	UKB	C	D	Explicit	Implicit	
<i>Pers. Type</i>	<b>PC</b>		-	-	-	+	-	-	+	-
	<b>RBP</b>	D	-	-	+	+	-	-	+	+
		C	-	+	+	-	-	-	+	-
	<b>VO</b>		-	-	+	-	-	-	+	-
	<b>R</b>	SA	-	-	-	+	+	-	-	+
		PA	+	-	+	-	-	-	+	-

Preference constructors (PC) are implemented in Preference SQL, and it is considered that the user would express the preferences explicitly by formulating queries. Here a utility and knowledge-based (*UKB*) method is being used in case when, for instance, a user states a certain attribute value in POS or NEG constructor and then preferences are propagated over all levels of the corresponding hierarchy (as seen in Example 2).

A content-based (*CB*) approach is applied in rule-based personalization (both RBP-D and RBP-C). In dynamic rule-based personalization (RBP-D), for instance, when ECA-rules are being executed, some information content is taken into consideration, e.g. the user role in the system as seen in Example 3. Also, a *UKB* approach is applied, when user behavior is being analyzed as a utility function calculates user's interest in certain aggregated data. In rule-based personalization with constraints (RBP-C) data is grouped in clusters as the result of the execution of some clustering algorithm (i.e. K-means clustering method), nonetheless, a user may influence data grouping by stating must-link and cannot-link constraints, thus, criticizing the proposed classification and experiencing a mixed initiative (*MI*) method. In RBP-D both implicit and explicit methods for collecting user information are applicable, whereas in RBP-C – only an explicit one.

A content-based approach is also employed in visual OLAP (*VO*) and in recommendations with user profile analysis (*R-PA*). In *VO* the user is capable of moving through the navigational schema and setting preferences for OLAP schema objects to be displayed – for example, one may choose dimensions, set constraints on dimension attribute values, etc. In *R-PA* a user states content-level preferences in the profile and ranks them with relevance scores. In both cases a user provides information explicitly.

In recommendations with user session analysis (*R-SA*) user information is gathered implicitly. To define user preferences, in [GMNS09, GMNS11, Mar14] authors present a *UKB* approach to exploit investigations in previous session queries, and apply a utility function conceptually similar to DIFF operator [Sar99]. In [KB12] to explore query logs and draw association rules between queries, a *UKB* approach is employed too. Both of these approaches produce multiple user recommendations, thus, are collaborative (*C*).

### 2.3. Summary of the Section

This section provides an overview of four directions of personalization in OLAP: preference constructors (*PC*), rule-based personalization (*RBP*) subdivided into dynamic rule-based personalization (*RBP-D*) and rule-based personalization with constraints (*RBP-C*), visual OLAP (*VO*), and recommendations (*R*) subdivided into recommendations with user session analysis (*R-SA*) and recommendations with user profile analysis (*R-PA*).

A comparative analysis was performed in order to point out (i) the level of personalization as well as personalization options described and its applicability to OLAP schema elements, aggregate functions, and OLAP operations, (ii) the type of constraints (hard, soft or other) used in each approach, (iii) the methods for obtaining user preferences and collecting user information.

One may observe that personalization of OLAP schema elements is mostly present in all proposed OLAP personalization types except for preference constructors (PC) and rule-based personalization with constraints (RBP-C). In PC the way of expressing user preferences for dimensions, hierarchies, fact tables as such as well as aggregate functions is not described. Out of all OLAP schema elements in RBP-C a user may only personalize hierarchies by adding a new customized hierarchy level (and, therefore, a new dimension attribute).

OLAP personalization approaches were characterized by the applied method for extracting user preferences [Bur02, VPF02]. The most widely-used methods are utility & knowledge-based and content-based; questions & answers, mixed initiative, and collaborative methods are applied each in one direction of personalization; demographic method is not employed.

There are two ways of gathering user preferences – either explicitly or implicitly. User preferences are collected explicitly in all methods except for R-SA, and implicitly – only in RBP-D and R-SA. However, in the papers considered in terms of this literature review the choice of the approach to gather user preference data is not well-grounded. Thus, one of the tasks of the practical study described in this thesis would include a comparison of methods that employ user preferences gathered either explicitly or implicitly to draw conclusions on which of the two approaches is more acceptable by users.

The aim of this literature study was to become aware of the existing state-of-the-art approaches in the field of data warehouse personalization and to determine a possible way of categorizing and characterizing them. It was important to understand, whether there is a gap in research and which of the approaches would be the most suitable for a new empirical study in the area of the data warehouse personalization.

Personalization 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 (*Common Warehouse Metamodel*, [CWM]) 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. The visualization of results plays a secondary role and often is restricted with

standard graphs. Such characteristics refer to both approaches that include query recommendations (R-SA and R-PA).

The OLAP reporting tool developed in the University of Latvia is CWM-based. It is a suitable experimental environment for introducing OLAP personalization by means of report recommendations. Though the OLAP reporting tool allows users to build their own reports, nevertheless, recommendations on query construction (as in [KB12]) 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.

In R-SA recommendations are created taking data as an input. In R-PA logical metadata (i.e. OLAP schema and its elements) serves only as auxiliary or context information and recommendations are still produced on data. The author of this thesis proposes to interpret user preferences as soft constraints, since soft constraints give more flexibility in providing results that reflect user interest. As the literature study shows, generation of report recommendations in a data warehouse reporting tool having OLAP schema and its elements as an input and interpreting user preferences as soft constraints is a subject for a new study, which is formally described in section 6 and makes up an original contribution of this thesis. It differs from other approaches involving query recommendations 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. One of the methods proposed in the thesis (see section 6.2.3.) 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. User preferences can be stated either implicitly or explicitly, since there is no common opinion on the superiority of any of these two approaches for gathering preferences.

The aim of the succeeding empirical study is to verify, whether metadata-based (or schema-specific) report recommendations can provide valuable guidance for exploration of the OLAP reporting tool regardless of user experience and familiarity with the data.



### **3. REQUIREMENT FORMALIZATION TO DEVELOP THE CONCEPTUAL MODEL OF A DATA WAREHOUSE IN COMPLIANCE WITH USER NEEDS**

#### **3.1. The Intent of the Section**

Apart from OLAP personalization opportunities (which in the context of this thesis cover report recommendations based on user preferences) to foster the delivery of the data of interest, the development of the conceptual model of a data warehouse that satisfies requirements is of high importance, too. First, a conceptual model of a data warehouse that complies with data warehouse requirements should be constructed, and afterwards OLAP personalization can be integrated. In terms of this thesis a research has been done that was targeted on the elaboration of the requirement formalization model to contribute to the development of the conceptual model of a data warehouse. The goal of the research study presented in this section is to use findings of the preceding research [NNK11, KNG13] in a real data warehouse project, which also includes a data mart employed in the experimentation with the reporting tool (see section 7), to extend the formal specification of indicators with elements discovered during the case study that is reflected in [KN14].

#### **3.2. Methods to Construct Conceptual Models for Data Warehouses**

A data warehouse stores data according to a multidimensional data model, which should be built in compliance with the analysis requirements of the organization. Therefore, one can speak about the information requirements [WS03]. Developing a data warehouse that fits all requirements of potential users is not the easiest task. Moreover, there is no common understanding about the best method for conceptual modeling of data warehouses and the most expressive modeling language for that purpose.

Conceptual models of data warehouses can be classified according to their origination [RALT06]: E/R model based, UML based, and independent conceptual models, e.g. Dimensional Fact Model [GMR98]. The necessity to develop special conceptual models for data warehouses is founded on existence of two types of data that should be modeled – quantifying and qualifying data, and elements of multidimensional paradigm, e.g. dimensions, hierarchies, cubes, whose semantics can't be modeled properly with standard modeling languages. Besides the specialized conceptual models for data warehouses, developers also need formal methods to construct these models [Riz09]. All methods can be classified as supply-driven or demand-driven according to how the data warehouse requirements are determined [WS03].

Supply-driven methods determine the existing information requirements during the analysis of data models of data sources, and in more or less automated way transform them into the data warehouse model. The limitation of supply-driven approach is that the constructed conceptual model may not reflect all analysis needs, because it reflects the operational needs of data source systems. A data warehouse model is obtained by transforming models of data sources, for example, [GMR98] analyze many-to-one associations in the data source models to construct an attribute tree that is used later to form dimensions, hierarchies, and other elements of multidimensional paradigm.

In demand-driven approaches the information needs are gained by interviewing users, therefore, the conceptual model of data warehouse depends on how precise the users formulate and data warehouse developers formalize the analysis needs. Precisely documented information requirements may serve further as a basis for semi-automated methods for development of a conceptual model of a data warehouse that afterwards can be checked for existence of source data. Demand-driven methods can be divided more accurately according to the way of identifying requirements, e.g. user-driven [Poe96, Wes01], process-driven [KO04], and goal-driven [LM04, GRG08] where users are interviewed and processes, goals, or indicators are modeled and analyzed to gain precise understanding of the analysis needs of users and the organization. For example, [Poe96] proposes a catalogue for storage of user interviews to collect end-user requirements, recommends to interview different groups of users to understand a business completely. In case of the process-driven approach, a business process is analyzed, e.g. in [KO04] the “as is” and “to be” process models are constructed including the analyzed processes, as well as the corresponding data models. In case of the goal-driven approach, goals of an enterprise, goals of business processes are analyzed and data that should be analyzed to achieve these goals is identified by means of filling in a number of templates during an interview. For instance, in [GRG08] the goal-oriented modeling is performed, facts and attributes are identified and mapped onto a conceptual model of a data warehouse, but hierarchies of each fact are later constructed by applying supply-driven approach [GMR98].

However, more approaches, e.g. ontology-based [RA10], pattern-based [JS05], are proposed to gain the most suitable conceptual model of a data warehouse for the implementation of the strategy of an organization and to avoid the limitations of existing methods.

An approach related to the demand-driven category was applied to build a conceptual model of the data warehouse of the University of Latvia, which includes a data mart used in the experimentation with the reporting tool (see section 7).

### 3.3. Existing Methods for Formalization of Data Warehouse Requirements

Let's recall the definition of a data warehouse [Inm02]: "A *data warehouse* is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management decisions". Hence, a data warehouse is a solution for data storage and analysis. In the context of a data warehouse one can consider the information requirements as different *indicators* that provide the basis for decision-making and supply the analyst with the necessary information. An *indicator* is a measure that is derived from other measures using an analysis model as measurement approach [GBC+06]. In its turn, an *analysis model* is an algorithm or a calculation combining one or more measures with associated decision criteria [GBC+06]. Indicators can be defined on various levels of formality. The definitions given in [GBC+06] are preferred over others, because the authors present an ontology that is aligned with different software measurement proposals, standards and metrology vocabulary.

During the elicitation of requirements for a data warehouse the information needs are expressed as more or less complex sentences in natural language that describe what data should be analyzed and how it can be measured. In this case, these sentences or requirements represent indicators. Typically, indicators formulated this way can be ambiguous or imprecise, thus, making it harder to interpret and reuse. The question is whether an indicator could be a subject of formalization – if the terms of the sentence could be structured and if a common pattern could be observed.

Some research on how to specify indicators has been done and is described in [PT07, FHKS08, PS10]. The authors of [FHKS08] propose a formal language for modeling goals based on performance indicators. Goal satisfaction could be controlled and evaluation of organizational performance could be performed. In [PT07] the authors propose a formal language for indicator definition by introducing the sorts of indicators, predicates and functions included in it. Relationships between indicators are defined. The authors claim that the usage of the considered specification language can be informal, semi-formal, graphical or formal. They argue that the requirements can be reformulated from natural language expressions to more formal. However, they do not use the formal representation of the indicator as an essential part of their specification language. In [PS10] the ideas of [PT07] are extended and the formalized indicators are integrated with other concepts, e.g. processes, goals, agents, etc. Indicator formalization [PT07] includes the definition of all relevant characteristics of indicators, e.g. name, definition, type, timeframe, etc.

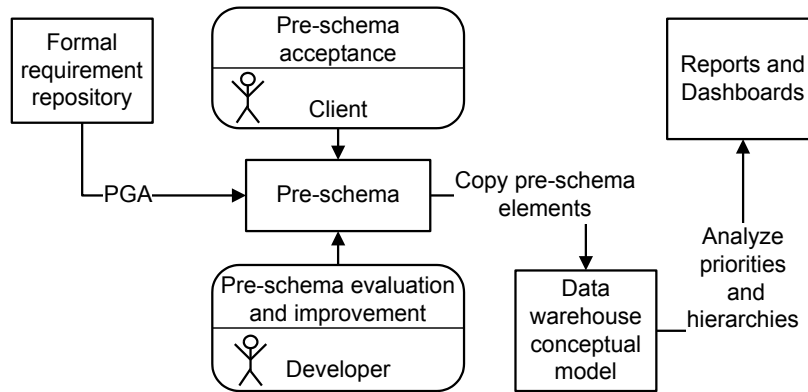
The authors of the research presented in [PAC+07] have based their proposal on User Requirements Notation (*URN*) [ITU03] which is a standard mostly used in

telecommunications systems and services, thus, URN application in the context of data warehouse requirements was not mentioned. It describes concepts that can be also applied in business process modeling including goal and requirement modeling. In [PAC+07] they exploit a data warehouse and integrate the Key Performance Indicators or KPIs concepts with the URN metamodel. *KPIs* represent the set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization [Par10]. Such indicator features as target value, threshold value, worst value, and others are defined.

The new approach presented in this section of the thesis formalizes not only different features of indicators like name, type, etc., but also tries to decompose the indicator definition in more detailed parts according to proposed indicator definition metamodel. Also, some features specific for the data warehouse development were added to the model, for instance, requirement priorities. The goal for such formalization is not only to describe the usage aspects of indicators and their dependencies, the formalized indicator definition can be used to semi-automatically generate a data warehouse model to store appropriate data that satisfies the indicators.

### **3.4. Requirement Formalization Metamodel and Examples**

Creating a repository of formal requirements is the first step of the method for transforming information requirements to the conceptual model of a data warehouse. In the author's paper [NNK11] a formal specification of indicators was proposed and a method to transform formally expressed information requirements or indicators into a conceptual model of a data warehouse was presented in the author's paper [KNG13]. The formal specification of indicators was built after analyzing a set of indicators from indicator database [Par10]. Since the semi-automated method for transforming information requirements to the conceptual model of a data warehouse has not been fully implemented yet and has not been evaluated, only the main components of the method are depicted in Figure 3.4.1 in this section. The author will mostly focus on the requirement formalization principles, description of the requirement formalization metamodel, and requirement examples in terms of this section. The requirement formalization metamodel may be employed to construct a conceptual model of a data warehouse independently of the semi-automated method. It helps to structure, systematize and evaluate requirements to give a better understanding of which of the requirements should have an impact on the development of the conceptual model of a data warehouse.



**Fig. 3.4.1.** Pre-schema generation and restructuring

The method uses a set of requirements, which are formalized according to requirement formalization metamodel and stored in the formal requirements repository, and generates a simplified data warehouse schema – a pre-schema – by the Pre-schema Generation Algorithm (PGA) that analyses the structure of requirements.

On the next stage of the method semi-automated pre-schemas are processed and restructured by developer to remove duplicates and build dimension hierarchies.

The improved schemas can be used as data warehouse schema metadata. All generated pre-schemas are being shown to the client during an interview, where the client should make a decision and choose one pre-schema that meets the requirements for a new schema best of all. The elements of the chosen pre-schema are being copied to the conceptual model of the data warehouse.

Finally, requirement priorities are propagated to schema elements to analyze, for example, which of the planned reports should be developed prior to others, which schema elements to incorporate into dashboards, etc.

A study that was mostly aimed at implementation of the interface for the formal requirement repository by means of Oracle APEX was reflected in bachelor thesis “Defining Formal Indicators to Develop the Conceptual Model of a Data Warehouse” (author – Dārta Liškauska, advisor Mg. sc. comp. Natālija Kozmina).

Let’s return to the part of the method for transforming information requirements to the conceptual model of a data warehouse (Figure 3.4.1) that is essential in the context of this thesis, i.e. the formal requirement repository with its underlying requirement formalization metamodel and other issues related to requirement formalization.

### 3.4.1. Principles of Requirement Reformulation

All of the principles of requirement reformulation mentioned in this section appeared from the practical experience after considering approximately 330 different indicators listed

in [Par10] and serve to translate the requirements from natural language to a state that is compatible with the requirement formalization model. The principles of requirement (indicator) reformulation are the following:

- A component to be measured is treated as an aggregated number of all occurrences of this component. For example, “sessions” is reformulated to “count (session)”, where “count” is the most suitable aggregate function;
- If an indicator component is supposed to be shown in detail, then in the corresponding requirement the refinement function “show” is applied. For example, “employee” is reformulated to “show employee”;
- If an indicator contains such components as “listing of”, “list of”, or “instances of”, then in the corresponding requirement the refinement function “show” is applied. For example, “listing of customers” is reformulated to “show customers”;
- If an indicator contains such component as “number of”, then in the corresponding requirement the aggregate function “count” is applied. For example, “number of visits” is reformulated to “count (visit)”;
- If an indicator contains such components as “cost of”, “value of”, “expense”, “total expense”, “income”, “total income”, “revenue”, “investment”, etc., or the name of currency in the beginning of the indicator, then in the corresponding requirement the aggregate function “sum” is applied. For example, “dollars saved” is reformulated to “sum (dollars)”, however, “total income” is reformulated to “sum (income)”;
- If an indicator contains such component as “average”, then the aggregate function “avg” is applied in the corresponding requirement. For example, “average response time” is reformulated to “avg (response time)”;
- If there are such components as “%”, “percent”, “percentage”, or “ratio”, then % is substituted by division of partial quantity by total quantity. For example, “IT expense as a % of total expense” is reformulated to “sum (IT expense) / sum (expense)”.

Of course, these principles are supposed to be used taking into consideration the context of each indicator. One should analyze indicators to decide whether the data has to be aggregated or not and choose the appropriate aggregate function, if needed. Some of the instances of such indicators are: sales closed, initiatives completed, dates, candidates, days of production, energy consumed, etc.

### 3.4.2. *Extending a Requirement Formalization Metamodel*

The initial version of the requirement formalization metamodel was published in [NNK11], and was tested (i.e. used to create formalized requirements) on approximately 330 different indicators (listed in [Par10]) from such measurement perspectives as customer focus, environment & community, employee satisfaction, finance, internal process, and learning & growth with the aim to check the compatibility of the metamodel with each of the indicators from the above-mentioned set.

The most complicated example of an indicator from this set would contain a ratio, for instance, a summary information on the percentage of IT expense of total administrative expense in a year would be formally written as “(sum (expense) where expense type = ‘IT’) / (sum (expense))”.

Since the initial version of the requirement formalization model was tested on a large set of indicators from the business sphere listed in [Par10], the goal of the following research study was to check the compatibility of the requirement formalization metamodel when applying it to a set of requirements for a real data warehouse project, and extend the formal specification of indicators with elements discovered in this case study. For that reason, the same requirement formalization metamodel has been tested on a set of requirements for the currently operating data warehouse of the University of Latvia. This data warehouse accumulates data on student enrolment statistics, student and academic staff activity in e-learning system, strategic indicators, staff workload statistics, etc. The overall number of requirements is over 150. While testing, it was stated that the metamodel should be extended with some additional classes like themes, grouping, and priorities, as well as relationships between classes should be reviewed.

Also, a small part of these requirements were more complex and consisted not only of ratios, but also an evaluation of these ratios (such as “the number of post-docs should increase by 10% by next year”), which led to extension and restructuring of the requirement formalization metamodel (a detailed explanation is given in section 3.4.3.).

### 3.4.3. *Two Versions of the Requirement Formalization Metamodel*

Since most of the information requirements have a common pattern, a metamodel to re-formulate these requirements in a formal way is applied. The metamodel is designed using UML class diagram notation (Figure 3.4.3.1). There are two versions of the requirement formalization metamodel – the initial version and the extended one depicted in Figure 3.4.3.1.

In the initial version of the requirement formalization metamodel [NNK11] a Requirement is classified either as Simple or Complex. A complex requirement is composed

of two or more requirements joined with an Arithmetical Operator. A simple requirement may consist of an Operation that denotes a command applied to an Object, and an optional Typified Condition. In its turn, an object is either an instance of Quantifying data (measurements) or Qualifying data (properties of measurements) depending on the requirement. A Complex Operation consists of two or more Actions, which are of two possible kinds: Aggregation (“roll-up”; for calculation and grouping) and Refinement (“drill-down”; for information selection). Information refinement is divided into showing details (selecting information about one or more objects), or showing details restricted with a constraint defined by Typified Condition (slicing). Just like requirements, Conditions and Expressions are either Simple or Complex. Complex condition joins two or more conditions with a Logical Operator. A simple condition, for instance, “year > 2013”, consists of a Comparison of two Expressions. A complex expression contains two or more expressions with an arithmetical operator in between, whereas a simple expression belongs either to qualifying data (e.g. “year”) or to Constants (e.g. “2013”).

The extended metamodel maintains all existing classes from the initial version. The results of the case study showed the necessity for certain improvements of the requirement formalization metamodel that are given below:

- Each requirement has to have its Priority value (must, should, could or won't). Benefits of adding priorities to requirements are explained in section 3.4.5;
- Two or more requirements, which make up a complex requirement as seen in example in Figure 3.4.4.1, may be joined with either an Arithmetical Operator or a Comparison (see a comparison between two complex requirements in Figure 3.4.4.1);
- A Simple Requirement should consist of an Operation that denotes a command applied to an Object, or of an Expression (see a constant = “10%” in Figure 3.4.4.1), and an optional Typified Condition. It allows to compare a part of the requirement with some expression or a pre-defined constant value, for instance, an informal requirement “The ratio of students to academic staff has to be 10.4” is reformulated to “(count (student) / count (academic staff)) = 10.4”;
- Each requirement may refer to one or multiple Groups (e.g. Dynamics, Master studies, Doctoral studies), whereas a Theme as a coarser level of grouping (e.g. Finance, Education, Customer Focus) may unite one or more groups. Grouping requirements is needed for several reasons: (i) to reduce the number of repeating elements in requirements, thus, making them more compact – for instance, if a number of requirements contains one and the same time frame (e.g. year), it can be added just once as a simple requirements “show year”; (ii) to unite multiple requirements



logically, which would be the natural grouping of reports to be developed later on (e.g. Dynamics, E-learning, Staff statistics, Student statistics).

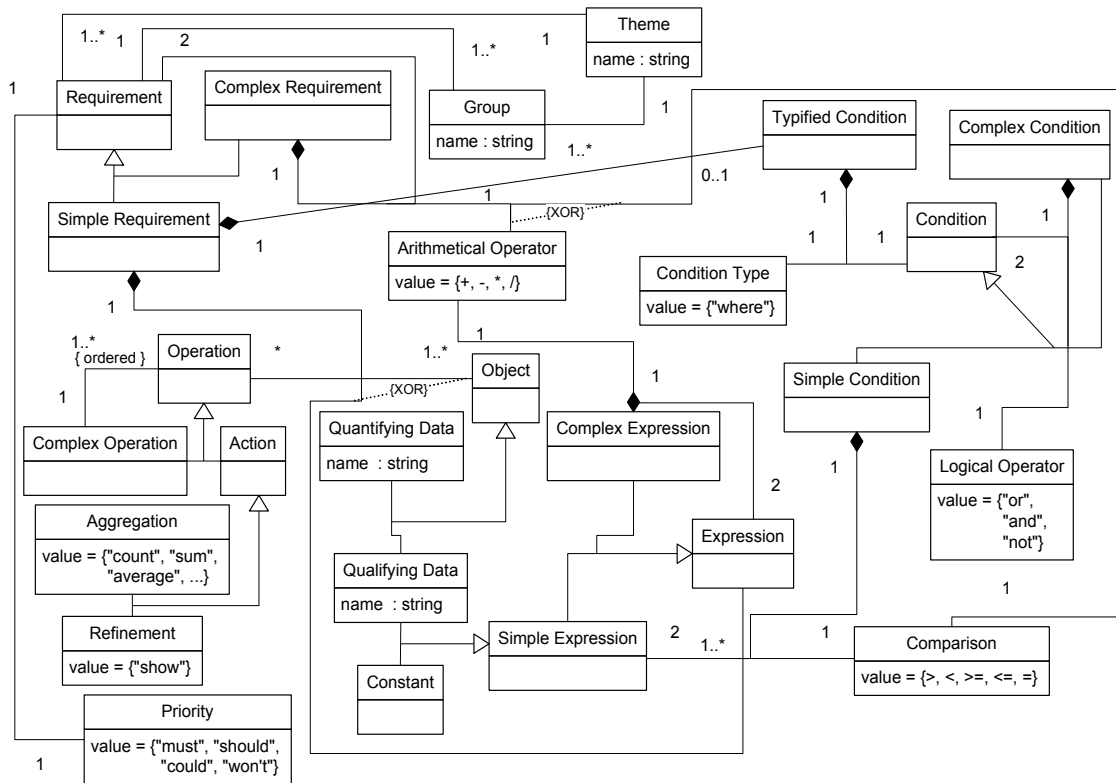


Fig. 3.4.3.1. An extended version of the requirement formalization metamodel [KN14]

### 3.4.4. An Example of a Formalized Requirement

Let's consider an example of a requirement formulated in compliance with the requirement formalization metamodel (Figure 3.4.4.1). Priority of the following requirement is "could", Theme is "Education", and Group is "Master studies". As these 3 classes solely characterize the requirement, but are not connected to other classes that help to form the requirement, they are excluded from the example in Figure 3.4.4.1.

Informally, the requirement goes as follows: "The ratio of master level graduates in the University of Latvia in 2013, who are employers, has to be 10% of master level graduates in the University of Latvia in 2012". In its turn, it is reformulated this way: "((count (graduate) where level = 'master' and year = '2013' and status = 'employer') / (count (graduate) where level = 'master' and year = '2013')) = (10% \* (count (graduate) where level = 'master' and year = '2012'))". "Has to" is interpreted as a request for equality, thus, these two complex requirements are linked with "=" sign.

count	Aggregation	Action	Operation		Simple Requirement	Typified Condition	Complex Requirement	Complex Requirement
graduate	Quantifying Data	Object						
where	Condition Type							
level	Qualifying Data	Simple Expression	Simple Condition	Complex Condition				
=	Comparison							
'master'	Constant	Simple Expression						
and	Logical Operator							
year	Qualifying Data	Simple Expression	Simple Condition					
=	Comparison							
'2013'	Constant	Simple Expression						
and	Logical Operator							
status	Qualifying Data	Simple Expression	Simple Condition					
=	Comparison							
'employer'	Constant	Simple Expression						
/	Arithmetical Operator							
count	Aggregation	Action	Operation		Simple Requirement	Typified Condition	Complex Requirement	
graduate	Quantifying Data	Object						
where	Condition Type							
level	Qualifying Data	Simple Expression	Simple Condition	Complex Condition				
=	Comparison							
'master'	Constant	Simple Expression						
and	Logical Operator							
year	Qualifying Data	Simple Expression	Simple Condition					
=	Comparison							
'2013'	Constant	Simple Expression						
=	Comparison							
10%	Constant	Simple Expression	Simple Requirement					
*	Arithmetical Operator							
count	Aggregation	Action	Operation					Simple Requirement
graduate	Quantifying Data	Object						
where	Condition Type							
level	Qualifying Data	Simple Expression	Simple Condition		Complex Condition			
=	Comparison							
'master'	Constant	Simple Expression						
and	Logical Operator							
year	Qualifying Data	Simple Expression	Simple Condition					
=	Comparison							
'2012'	Constant	Simple Expression						

Fig. 3.4.4.1. An example of a formalized requirement

The left column is filled with parts of the requirement statement and all the rest columns (left to right) contain class names of the requirement formalization metamodel. This requirement has a sophisticated structure and it is a complex requirement that consists of two

others. In the requirement example given in Figure 3.4.4.1 two principles of requirement (indicator) reformulation (see section 3.4.1.) are applied:

- A component to be measured is treated as an aggregated number of all occurrences of this component: “graduates” is reformulated to “count (graduate)”, where “count” is the most suitable aggregate function;
- If there are such components as “%”, “percent”, “percentage”, or “ratio”, then % is substituted by division of partial quantity by total quantity: “ratio of master level graduates in the University of Latvia in 2013, who are employers” is reformulated to “((count (graduate) where level = ‘master’ and year = ‘2013’ and status = ‘employer’) / (count (graduate) where level = ‘master’ and year = ‘2013’))”.

### 3.4.5. Requirement Prioritization

One of the classes in the extended version of the requirement formalization metamodel is Priority. Involving a client in setting priorities at the stage of requirement elicitation adds value to the process of construction of the conceptual model and following report development. For instance, to be more specific, translating requirement priorities to schema elements allows to estimate which of the pre-schemas (see Figure 3.4.1) is better aligned with high-priority requirements. This and other aspects of requirement prioritization will be discussed further.

The requirement prioritization technique chosen and integrated into the requirement formalization metamodel (see Figure 3.4.3.1) is *MoSCoW* analysis described in Business Analysis Body Of Knowledge (BABOK) Guide [B09]. This is a fast and straightforward approach that doesn't require complex calculations during re-prioritisation process and works best for assigning priorities in small groups of decision-makers (1-5 people). In MoSCoW analysis requirements are divided into four groups: must, should, could, and won't, which are defined in [B09] as follows:

- “Must” describes a requirement that must be satisfied in the final solution for the solution to be considered a success;
- “Should” represents a high-priority item that should be included in the solution if it is possible; this is often a critical requirement but one that can be satisfied in other ways if strictly necessary;
- “Could” describes a requirement, which is considered desirable but not necessary, and will be included if time and resources permit;
- “Won't” represents a requirement that stakeholders have agreed will not be implemented in a given release, but may be considered for the future.

Requirement priorities may be redefined when needed, however, it is advised to consider the proportion of maximum total effort: must – 60%, should – 20%, could – 20% (won't requirements are not included into it). MoSCoW analysis works best when priorities are discussed and assigned in groups.

As requirement elements are tightly connected to pre-schema (see Figure 3.4.1) elements, setting priorities to requirements would help to answer the following questions:

- How requirement priority values are propagated to schema elements?
- Which pre-schema is most likely to be accepted by client?
- Which elements of the accepted pre-schema to incorporate into dashboards?
- Which of the planned reports should be developed prior to others?

Let's take a look at each of the above-mentioned points in more detail.

### ***How requirement priority values are propagated to schema elements?***

A method of transforming requirements to the conceptual model of a data warehouse (pre-schemas) is described in [KNG13]. A pre-schema generation algorithm (PGA) is employed for distinguishing data warehouse schema elements in formalized requirements, which are stored in formal requirement repository. Thus, if there is some requirement R with a certain priority P, then all schema elements derived from the requirement R (i.e. measures and attributes) have their priority value set to P. Imagine that one and the same schema element (e.g. a Study Program attribute) has more than one priority value (e.g. must, could) gained from a set of requirements with various priorities. If a schema element has multiple priority values, then the one of the higher value is assigned (e.g. a Study Program attribute is assigned a "must" priority value).

### ***Which pre-schema is most likely to be accepted by client?***

A pre-schema, which includes the largest number of schema elements corresponding to components of requirements with higher priority (i.e. must, should), is the one that is most preferred by the client. There may be more than one way to evaluate each pre-schema; however, the most natural way is to count schema elements of each priority value and sort the acquired 4 values by priorities (must, should, could, and won't) in descending order. Thus, one may obtain a sorted list of pre-schemas based on requirement priorities.

### ***Which elements of the accepted pre-schema to incorporate into dashboards?***

A *dashboards* provides an interactive summary of data by organizing multiple reports into a single layout. Dashboards often demanded by decision-makers should not be

overwhelmed with data. Only the most essential reports are represented in dashboards. Therefore, the goal is (i) to detect elements of the accepted pre-schema with highest priorities from the corresponding requirements, and afterwards (ii) to check if any of these elements build up data hierarchies.

The next step would be the creation of requirement hierarchies based on hierarchies in schema elements. The PGA described in [KNG13] may determine attributes and measures from requirement objects, i.e. qualifying and quantifying data respectively. It means that one can analyze requirements that contain the same quantifying data (corresponding to measures) and typified conditions, but different qualifying data (corresponding to attributes).

Suppose, there is a pair of (already formalized) requirements such as:

*R1*: show course count (user session) where user role = “student”

*R2*: show course category count (user session) where user role = “student”

Consider a Course hierarchy in Course dimension: Course  $\rightarrow$  Course Category. Here “user session” in e-learning system is related to a measure, whereas “course” and “course category” are related to attributes. In this case, *R1*  $\rightarrow$  *R2* is a requirement hierarchy example, because corresponding schema elements form a hierarchy. A dashboard report in this case would be the one based on *R2* requirement.

Finally, in a given pre-schema those schema elements that are related to the requirements of the coarser level of granularity (e.g. *R2*) with highest priority are selected and treated as components of a potential report for a dashboard. Dashboard reports may be explored more in-depth sliding down to finer levels of granularity of one or another axis.

### ***Which of the planned reports should be developed prior to others?***

It is worthy to notice that the structure of a formalized requirement (i.e. the one that includes qualifying and/or quantifying data with or without additional restrictions) is such that it allows to build a data warehouse report containing schema elements that correspond to qualifying and quantifying data in requirements. In other words, it is quite an easy task to define the potential reports out of initial requirements stated by client.

Thus, having split all the requirements into 4 groups – i.e. must, should, could, won't – it is possible to create exactly 4 groups of labels for reports respectively in the context of time – namely, most urgent, urgent, less urgent, not urgent. This approach would help to sort the report that should be created prior to others.

### **3.5. Summary of the Section**

In this section a problem of delivering a conceptual model of a data warehouse that is in line with the client's needs was tackled. The quality of the conceptual model has an impact on further data warehouse personalization as such. If the conceptual model does not fully reflect needs of a client, then neither will OLAP personalization do it.

The requirement formalization metamodel is the initial step of the methodology for transforming requirements into a conceptual model of the data warehouse [KNG13]. This metamodel is necessary for creating a formal requirement repository out of information requirements in natural language, and it was reviewed in terms of this section. The research results described in this section are published in the paper [KN14].

A case study was conducted that consisted of testing the existing requirement formalization metamodel, i.e. the findings in the preceding research [NNK11], on a set of requirements for a real currently operating data warehouse project of the University of Latvia. These requirements related to student enrolment statistics, student and academic staff activity in e-learning system, strategic indicators, staff workload statistics, etc. The overall number of requirements was over 150. Due to a specific structure of requirements that contain an evaluation of ratios (such as "the number of post-docs should increase by 10% by next year"), it was stated that the metamodel had to be restructured and extended with some additional classes like themes, grouping, and requirement priorities, as well as relationships between classes had to be reviewed.

Having chosen MoSCoW analysis as the most suitable requirement prioritization technique, the following questions were addressed: (i) which of the planned reports should be developed prior to others, (ii) how requirement priority values are propagated to schema elements, (iii) which schema elements to incorporate into dashboards, and (iv) which pre-schema is most likely to be accepted by client.

The risk of interpreting information requirement erroneously is threefold: a client might be imprecise in formulating the needs, an interviewer might capture them incorrectly, and, finally, a developer might construct a conceptual model that does not fully comply with information requirements stated by the client. The requirement formalization metamodel serves to minimize the risk at all three stages and to ensure that the conceptual model of a data warehouse is aligned with the information requirements. Additionally, requirements can be formalized independently of the PGA algorithm as it was done while constructing a set of reports to prepare the experimental environment for the research in the field of OLAP personalization described further in the thesis.

## 4. USER-DESCRIBING PROFILES IN OLAP

### 4.1. The Intent of the Section

The intent of this section is to propose a model that describes a data warehouse user with a set of generic profiles in order to cover various aspects of OLAP personalization and to structure user-describing attributes, some of which will be employed to personalize the reporting tool. The basic idea of the development of user-describing profiles is inherited from the Zachman Framework concept [Zac, Zac03].

### 4.2. The Concept of User-describing Profiles

Zachman Framework is an ontology that allows describing an arbitrary object from different viewpoints (temporary, spatial, and other aspects). Zachman Framework concept is used to give detailed characteristics of data warehouse user interaction with the system environment. To identify and develop profiles, the following questions were used: *who*, *what*, *how*, *when*, *where*, and *why*. A detailed representation of user-describing profiles is provided in Table 4.

*Table 4. User-describing profiles*

<b>Question</b>	<b>Description</b>	<b>Profile Type</b>
<i>Who</i> is the user?	Basic user data (personal data, session, activity, rights, etc.)	User
<i>Where</i> is the user located?	User physical location data & geolocation according to user IP-address	Spatial
<i>When</i> does the user interact with the system?	Time characteristics of user activities	Temporal
<i>How</i> does the user & system interaction happen?	Characteristics of user device (i.e. PC, laptop, mobile phone, etc.), which is used for signing in as well as user software (e.g. web browser) characteristics	Interaction
<i>What</i> is the user expecting to get as a result?	User preferences data	Preferential
<i>Why</i> the user is interested in this particular system?	User preferences are being gathered and analyzed; recommendations are generated, according to user characteristics and preferences	Recommendational

Similar method has been applied in the field of data warehouses by [JS05]. Here the authors present the Dimensional Design Patterns (DDPs) that would assist in designing the conceptual model of a data warehouse by providing an approach for identifying dimensions in

a systematic way. Thus, DDPs [JS05] are design-oriented, whereas user-describing profiles are user-oriented.

The proposed profiles describe user environment, i.e. different aspects of data warehouse user interaction with the system. User, spatial, temporal, interaction, and preferential profiles altogether compose a versatile description of the data warehouse user.

### 4.3. The Method for Construction of User-describing Profiles

The method suggested in this section consists of the following steps:

1. Stating questions (what? who? how? etc.) to enable the description of data warehouse user/system interaction;
2. Identifying the user describing profiles;
3. Collecting possible attributes of user-describing profiles from various sources of information (see Table 5);
4. Generating user characteristics via profile attributes;
5. Suggesting possible recommendations for novice and experienced users of reporting tool based on report preferences for the contents and structure of reports (OLAP preferences) and visual layout preferences;
6. Selecting recommendations from *TopN* recommendation list.

User, interaction, temporal and spatial profiles consist of attributes that describe the user. To construct sets of attributes for each of the mentioned profiles, a certain method has been applied.

**Table 5. Information sources of the user profile attributes (fragment)**

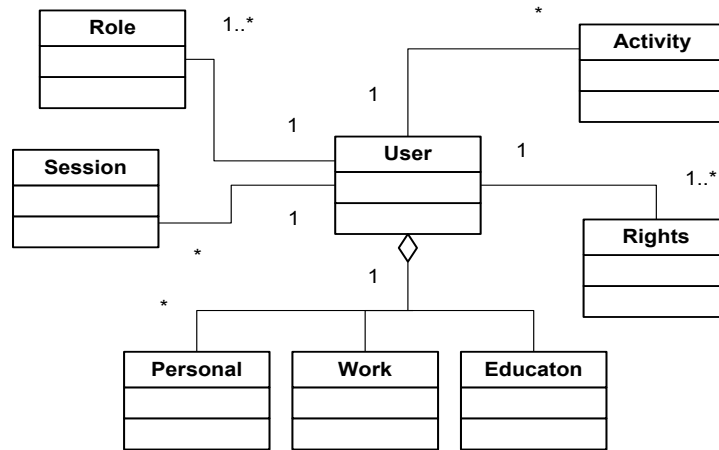
<b>User Profile Attributes</b>	<b>Information Sources</b>
Salutation, FirstName, LastName	[KR02, Sil01, PCTM03]
InformalGreetingName, FormalGreetingName, Suffix, Ethnicity	[KR02]
Gender	[Sil01, PCTM03]
Username, Citizenship, BirthDate, MaritalStatus	[Sil01]
Residence, AgeGroup	[IGG03]
...	...

The method for profile construction includes studying of data warehouse literature (e.g. [KR02, Sil01, JKP04]), CWM standard (*Common Warehouse Metamodel*, [CWM, PCTM03]), scientific and technical articles (e.g. [MTL, IGG03]), as well as practical experience in data warehouse field and working with data warehouse tools (e.g. Oracle



Warehouse Builder) and web-services (e.g. [IPAG, MBI, FIA]). User-describing profiles have been built by means of collecting various attributes from different information sources (see Table 5).

An attribute set of each profile has been logically split into classes in order to compose a class diagram for each user-describing profile. A class diagram of the user profile is depicted in Figure 4.3.1., however, attributes of the user profile classes are omitted. A description of classes of each user-describing profile will follow. Each class may be complemented with more attributes, if necessary. In terms of this section only class diagrams of user and preferential profiles are presented, since the elements of these profiles will be extensively exploited in OLAP personalization study further in the thesis, in its turn, class diagrams with attributes of all the profiles are available in the author's master thesis [Koz10].



**Fig. 4.3.1.** User profile class diagram

User profile classes:

- *Role* – contains the user system role attribute,
- *Personal* – contains 28 user personal information attributes (e.g. first name, last name, gender, ethnicity, marital status, age group, current passport nr.),
- *Work* – contains 25 attributes describing user work (e.g. position, company name, total years of experience, business trip day count per year),
- *Education* – contains 11 attributes describing user education (e.g. currently student, educational institution, year of graduation, diploma nr., honors),
- *Session* – contains 9 attributes describing user session characteristics (e.g. session start, session length, success status, session type, session context),
- *Activity* – contains 4 attributes indicating user activity (e.g. hit count & spent time) on a certain webpage in a certain period of time (e.g. full date),

- *Rights* – contains 7 attributed describing user rights for certain objects (e.g. table, column) of a reporting tool (e.g. can read, can edit, can delete).

Temporal profile classes:

- *StandardCalendar* – contains 22 standard calendar attributes (e.g. day number in month, month abbreviated, month number in year),
- *FiscalCalendar* – contains 12 fiscal calendar attributes (e.g. fiscal convention, fiscal week, fiscal year start date, fiscal quarter),
- *Time* – contains 7 non-calendar attributes and attributes that represent a date as a number (e.g. hour, SQL date stamp, seconds since midnight, Julian date),
- *TimeStatus* – contains 12 attributes of yes/no type (e.g. holiday, weekend, last year in month, peak period),
- *DomainSpecific* – contains 13 attributes specific for one or another domain (e.g. time-characterizing attributes of educational domain are semester, acad. year),
- *SpecialPeriod* – contains 7 attributes that describe certain planned or spontaneous global or local events (e.g. selling season, local special event – for instance, short-term strike, or global special event – for instance, earthquake or volcano eruption).

Spatial profile classes:

- *PhysicalLocation* – contains 22 attributes describing person's physical address (e.g. street name, street direction, suite, countryside, city, country),
- *LocationByIP* – contains 14 attributes derivable from user IP-address by means of web-services (e.g. postal code, time zone, continent, latitude, longitude).

Interaction profile classes:

- *WebAccess* – contains 15 attributes describing operating system, web-browser, and Internet connection properties (e.g. connection speed),
- *Functional* – contains 26 attributes describing web-browser functional properties and supported applications (e.g. AdobeAcrobat, Quicktime),
- *VisualLayout* – contains 12 attributes describing visual layout properties in a web-browser (e.g. color depth, browser dimensions, font smoothing, font sizing)

Construction methods of preferential and recommendational profiles differ from that previously described. While stating preferences, the user is able to select attributes from user, interaction, temporal, and spatial profiles. Multiple scenarios, which describe user preference types, have been considered, while constructing preferential profile (see the examples of

scenarios in section 5.2.5.). Recommendation profile (see section 4.3.3) contains recommendations based on preferences that belong to different users or a single user (i.e. individual recommendations). In its turn, a *recommendation* in the context of the reporting tool is a link to another report that matches user preferences.

#### 4.3.1. User-describing Profile Connections and Data Sources

One user may have more than one spatial, temporal, interactional, preferential, or recommendational profile. User-describing profile connections are depicted in Figure 4.3.1.1. For instance, signing into the system using PC or mobile phone leads to construction of two separate interaction profiles belonging to one certain user that contain different data about the device screen resolution. Thus, the diversity of user-describing profiles gives an opportunity to apply personalization adjusting the report structure, its visual layout, and its contents according to the data in user-describing profiles.

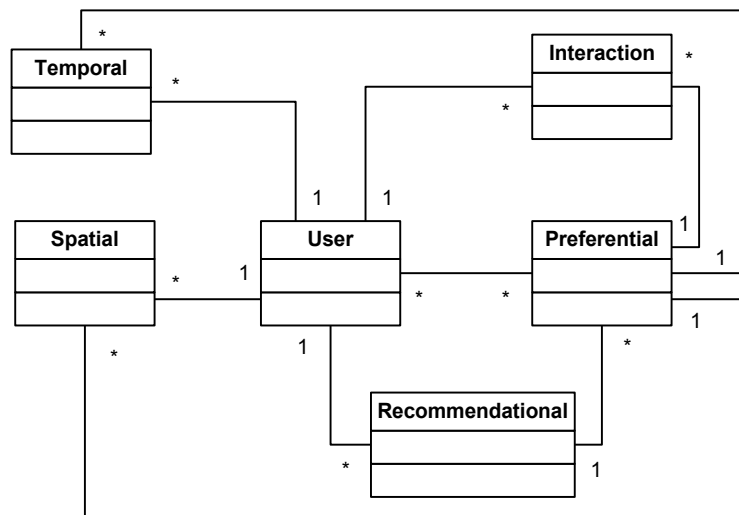


Fig. 4.3.1.1. User-describing profile connections

Preferential profile is connected with temporal, spatial, user, and interaction profiles, because the user may state his/her preferences on attributes of the mentioned profiles.

Recommendational profile contains recommendations based on sets of user preferences belonging to either multiple users or a single user. Recommendations come in handy when a user is not determined about the reports he/she might be interested in.

A single profile may contain many attributes with values assigned. However, there are multiple data sources to collect the profile attributes from; these data sources are shown in Figure 4.3.1.2. Let's consider these data source.

*Context* data (i.e. device used, operating system, IP-address, web-browser, etc.) describes the environment, in which the reporting tool is being employed. Context data is

gathered automatically by means of web-services [IPAG, MBI, FIA]. All values of the interaction profile attributes refer to context data as well as a part of the spatial profile attributes (i.e. geolocation by IP-address).

*Static* data is gathered from the values of the dimension attributes of a data warehouse. All values of the temporal profile attributes, a part of the spatial, and values of the user profile attributes are static.

*Activity* data is derivable from the data warehouse log-tables. In the user profile, activity data indicates the intensity of the reporting tool usage defined by the user hit count and time spent.

*Analysis* data refers to recommendational profile as recommendations are generated after analysis of user preferences.

*Explicitly entered* data is the data entered by a user manually. All values of the preferential profile attributes, which indicate the importance of one or another user preference (i.e. degree of interest, weight or priority), are gathered from a user explicitly. It is demonstrated in Figure 4.3.1.2 that explicitly entered data is acceptable in interaction, spatial, temporal, and user profiles, because the user can enter and/or edit attribute values of the mentioned profiles.

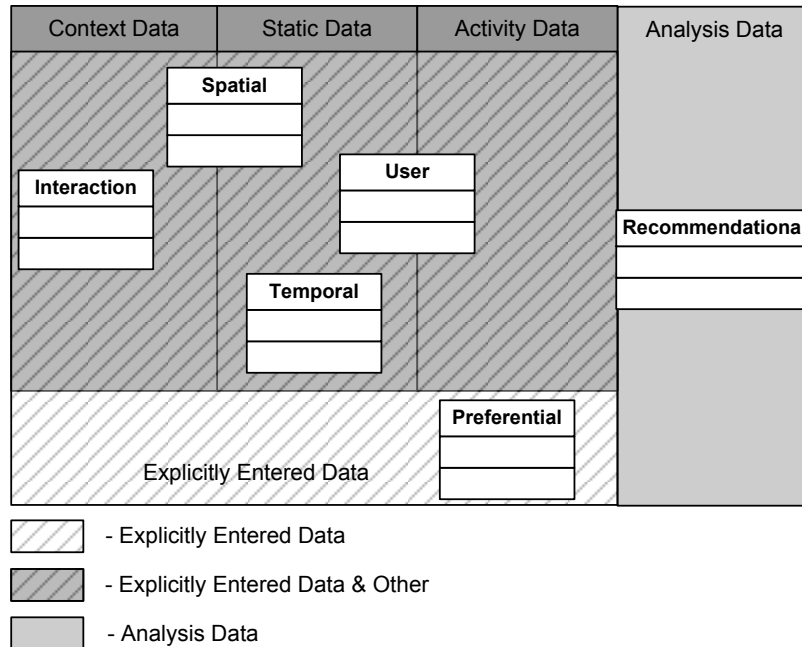


Fig. 4.3.1.2. User-describing profile data sources

#### 4.3.2. A Concept of the Preferential Profile

Before developing the user preference metamodel, which is presented further in the thesis, it was important to classify user preferences for reports. To reach this goal, various

user preference modeling scenarios have been considered, which later have been divided into two groups:

- Preferences for the contents and structure of reports (OLAP preferences),
- Visual layout preferences.

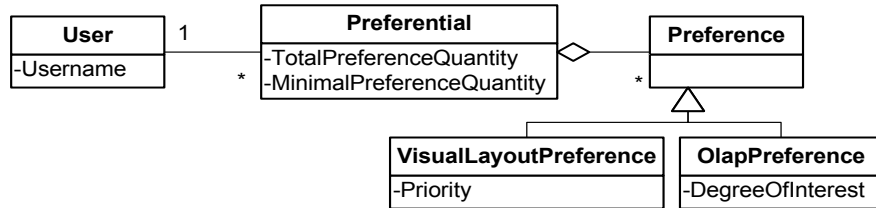


Fig. 4.3.2.1. Preferential profile metamodel (generalized)

Although, user preference metamodel contains two distinct classes of preferences (Figure 4.3.2.1.) – OLAP and Visual layout (Figure 4.3.2.2.) – in terms of this thesis, only methods that operate with OLAP preferences are implemented in the experimental environment (i.e. OLAP reporting tool); visual layout preferences are omitted, because they are of lower priority.

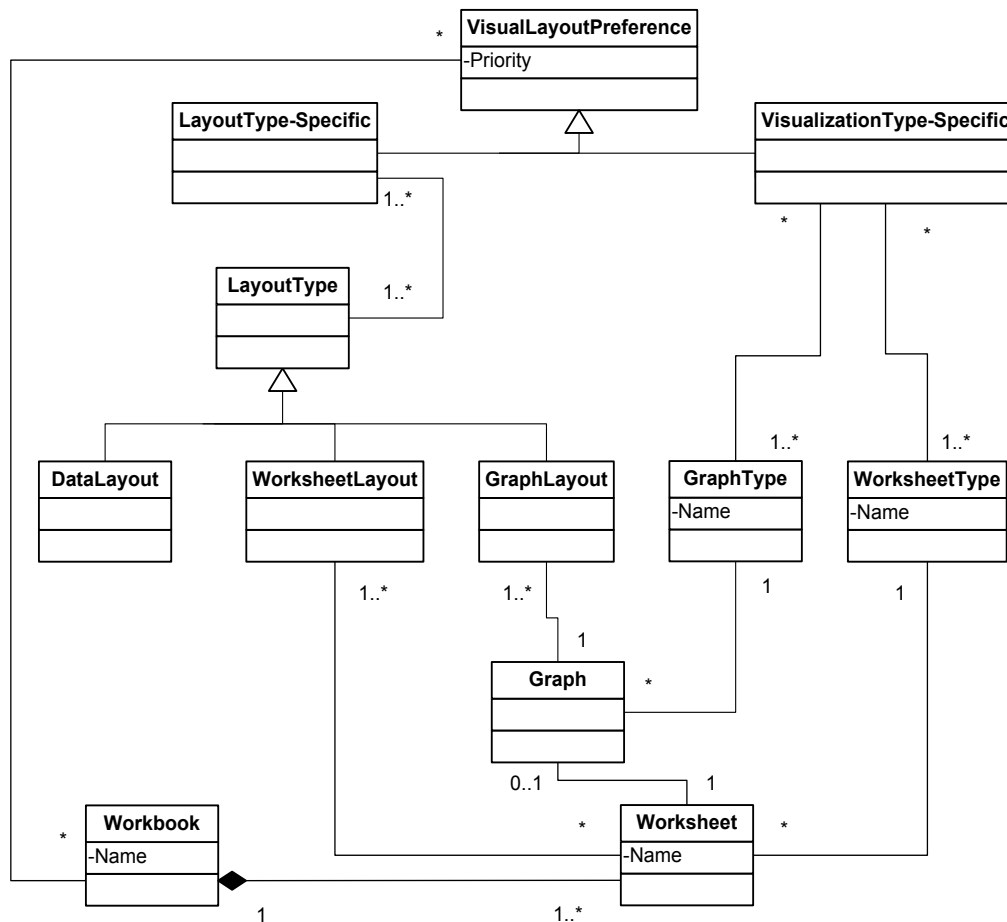


Fig. 4.3.2.2. Visual layout preferences metadata

Priority attribute of the class `VisualLayoutPreference` is a non-negative natural number and denotes the importance of each preference – the higher is the priority value, the more important is the visual layout preference. Thus, Priority attribute allows to define the order of visual layout preferences to be applied.

The scope of visual layout preferences is a set of reports or worksheet (i.e. workbook). Visual layout preferences split into layout-specific (`LayoutType-Specific`) and visualization-specific (`VisualizationType-Specific`). Each layout-specific preference may contain one to many layout type (`LayoutType`) elements. `LayoutType` class includes three subclasses: `DataLayout` class to describe visual layout of report data, `WorksheetLayout` to describe visual layout of report itself, and `GraphLayout` to describe visual layout of a graph. Each report may have not more than one graph to display report data with at least one graph visualisation type (`GraphType`). Each report may have at least one its visualization type (`WorksheetType`).

A detailed description of user OLAP preferences and OLAP preference metamodel is put forward in a separate section (see section 5.2.5.).

### 4.3.3. *A Concept of the Recommendation Profile*

Sometimes a user has no idea about what kind of data he/she is able to find in data warehouse reports. In this case the preferential profile is employed to produce recommendations, which altogether make up a user's recommendational profile and are calculated on the basis of either (i) a preferential profile of a single user or (ii) preferential profiles of multiple users that have something in common. The latter approach is widely used in recommender systems.

Recommender systems operate with such entities as users and items. A user of the recommender system expresses his/her interest in a certain item by assigning a rating (i.e. a numeric equivalent of user's attitude towards the item within a specific numerical scale). In [VM03, DHK08] an overview and analysis of methods employed in recommender systems is presented. One may distinguish user-based, item-based, and hybrid methods that combine principles of user-based and item-based ones [Bur07].

A considered user (or item) is referred as an *active* one in order to be distinguished from all other users (or items) of the recommender system.

User-based methods refer to collaborative filtering and user-based k-NN algorithm (introduced by [RIS94]). These methods would work best for multiple users whose preferential profiles have something in common. The similarities between each pair of users are calculated according to the ratings given to common items that both users have expressed their opinion on. Then, the neighborhood is formed around the active user, which consists of

users with the closest similarity values to the one of the active user. Prediction on the item value is made taking into account ratings of neighborhood users on the same item.

Item-based methods refer to content-based filtering and involve the item-based k-NN algorithm (introduced by [SKKR01]). These methods produce individual recommendations and are suitable for the preferential profile of a single user. The similarities are calculated for each pair of items rated by a common user. Active item's predicted value may be computed by means of weighted average of ratings on similar items.

Typically, to limit the number of recommendations such filtering criterion as *TopN* is applied in recommender systems, which means that only  $N$  recommendations will be shown to a user. Besides,  $N$  is either a fixed numeric value (e.g. 5, 10) or may be defined arbitrarily by a user.

#### 4.4. Summary of the Section

In this section a new method has been proposed, which provides an exhaustive description of interaction between a user and a data warehouse employing the concept of Zachman Framework [Zac, Zac03] according to which a set of generic user-describing profiles (user, interaction, temporal, spatial, preferential, and recommendational) has been developed.

A model that reflects connections among user-describing profiles and a diagram that characterizes profile data sources has been proposed. To construct sets of attributes of user, interaction, temporal, and spatial profiles, literature studies have been performed. As a result, class diagrams for user, interaction, temporal, and spatial profiles have been developed. Recommendational profile contains recommendations calculated on the basis of (i) a preferential profile of a single user or (ii) preferential profiles of multiple users that have some common preferences.

In this thesis special attention is paid to the 5th and the 6th steps of the method (see section 4.3.), namely, suggesting possible recommendations (organized in *TopN* lists) for novice and experienced users of the new OLAP reporting tool based on their preferences collected in preferential profiles. Nevertheless, there are some limitations that apply to further sections:

- The methods proposed by the author of the thesis that employ OLAP preferences to generate recommendations in the new reporting tool exploit schema-specific OLAP preferences only (see section 5.2.5) due to the lack of research results on the methods for generating recommendations on the basis of OLAP schema elements (for more details see section 2);

- Recommendations in the reporting tool are generated individually for each user taking as an input his/her preferences only. It is done this way, because users of the reporting tool might have different rights on reports. Thus, recommendations generated on the basis of preferences of the group of users might be of little help to a certain user, because he/she doesn't have the rights to execute some report(s) from *TopN* recommendation list.



## 5. OLAP REPORTING TOOL AND ITS METADATA

### 5.1. The Intent of the Section

A thorough practical experience of the author of this thesis with commercial tools for report design (e.g. Oracle Discoverer, MicroStrategy Analytics) showed that personalization is often limited by visual appearance of the report. In case of MicroStrategy, after executing a report there is an option to 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.

OLAP reporting tool developed at the University of Latvia is considered as an experimental environment for introducing OLAP personalization. The reporting tool is a part of the data warehouse framework [Sol07]. All operation of the data warehouse framework and the reporting tool as a part of it is based on metadata that is used to describe data warehouse schemas, their storage in relational database, and semantics of data stored in a data warehouse as well as to accumulate information about reports defined by users on data warehouse schemas. The intent of this section is to provide technical details on the implemented OLAP reporting tool and to introduce its metadata that consists of five interconnected layers: logical, physical, reporting, semantic, and OLAP preferences metadata.

### 5.2. Metadata Layers

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 (see Figure 5.2.1).

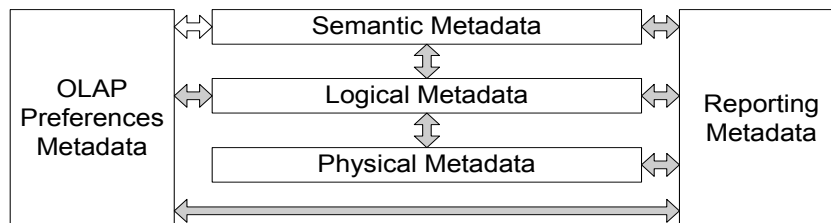


Fig. 5.2.1. Connections of metadata layers in OLAP reporting tool

Logical metadata is used to describe data warehouse schemata (or schemas). 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 schemas. OLAP preferences metadata stores definitions of user preferences on report structure and data.

Particular classes these metadata layers are connected by associations. Semantic metadata describes report 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 report structure are OLAP schema elements from the logical metadata that correspond to concepts from the semantic metadata, and components of user preferences on report 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.

CWM or *Common Warehouse Metamodel* [CWM] was used as a basis for the physical, logical, and semantic metadata, and supplemented with several new classes. Physical, logical, and semantic metadata layers are described in sections 5.2.1., 5.2.2., and 5.2.4. respectively.

### **5.2.1. *Physical Metadata***

CWM contains a package Relational, which was taken as a basis for physical metadata (Figure 5.2.1.1). It describes relational database schema of a data warehouse and the mapping of a multidimensional schema to relational database objects. The physical metadata [Sol08b, Sol10] is connected to the logical metadata by mappings of attributes and measures to one or several columns.

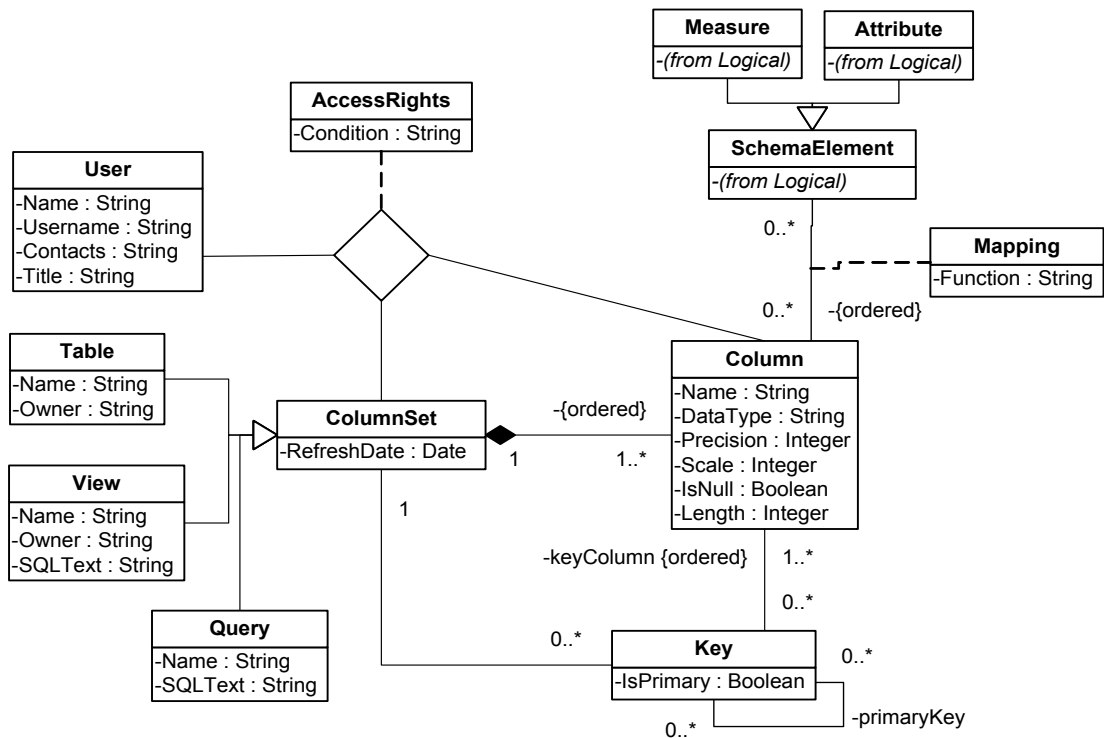


Fig. 5.2.1.1. Physical metadata [Sol08b]

### 5.2.2. Logical Metadata

Metadata at the logical level describes the multidimensional data warehouse schema (Figure 5.2.2.1).

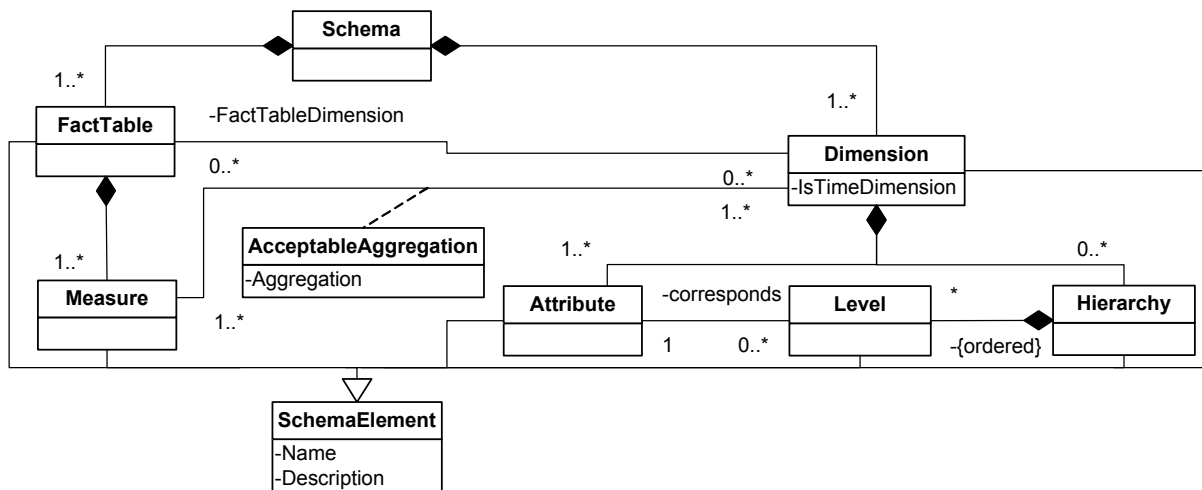


Fig. 5.2.2.1. Logical metadata [Sol08b]

The logical level metadata is based on the OLAP package of CWM and contains the main objects from this package such as dimensions with attributes and hierarchies, fact tables (cubes in CWM) with measures. FactTableDimension associations connect fact tables and dimensions. Only dimensions and fact tables connected by FactTableDimension associations

can be included together in one report. 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 papers [Sol08b, Sol10].

According to the logical level metamodel, data warehouse schema elements (class `SchemaElement`) are included into a hierarchical structure: a data warehouse schema is composed of interconnected fact tables and dimensions, which are composed of measures and attributes respectively. Dimensions also include hierarchies composed of ordered levels defined by attributes. A fact table belongs to exactly one schema, but a dimension can be shared among multiple schemas.

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

### **5.2.3. Reporting Metadata**

Reporting metadata describes the structure of reports on data warehouse elements (Figure 5.2.3.1). 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) grouped in the class `SchemaElement`. Reports also consist of user-defined conditions and joins between tables.

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 reports, 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 OLAP reporting tool reports are defined by developers or by experienced users themselves by means of choosing the desired elements of a data warehouse schema and defining conditions, parameters, etc. To define a report, users are allowed to select measures and attributes belonging to one schema. 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 the reporting metadata [Sol08a, Sol10], and its result is displayed to a user.

In this thesis only the 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 the structure. The information about item visibility is obtained from the attribute Location of the class Item in the reporting metadata.

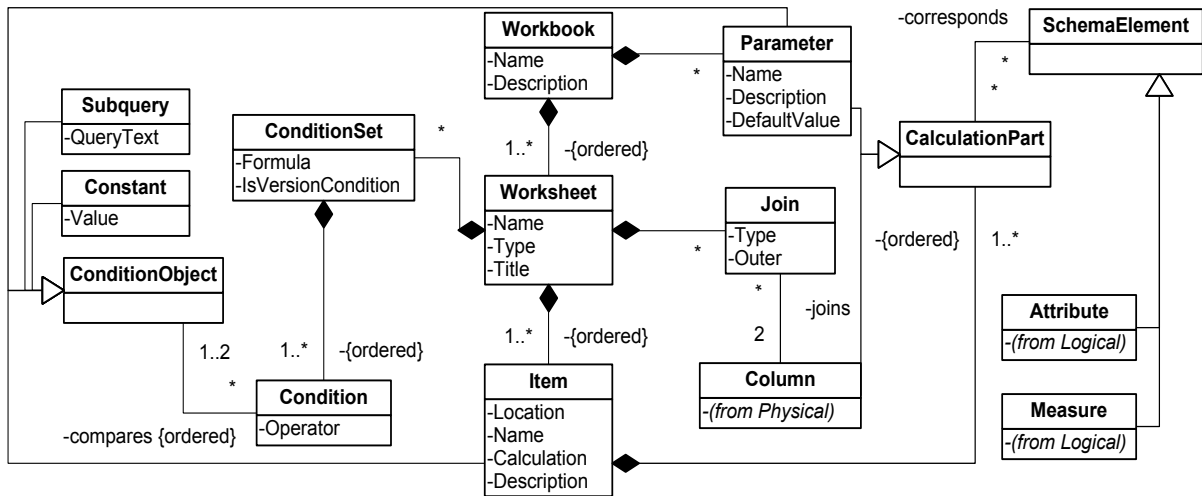


Fig. 5.2.3.1. Reporting metadata [Sol08a]

### Interconnection of Report Items and OLAP Schema Elements

The models of logical level metadata and reporting metadata are interrelated. Report items are defined by computation formulas from calculation parts that correspond to table columns or parameters. If a calculation part corresponds to a certain dimension attribute or measure, then this schema element is connected to the class CalculationPart by the association 'corresponds' in the reporting metadata. Data warehouse schema elements (measures and attributes) that were used to calculate report items are determined according to the correspondence associations between calculation parts of the item and schema elements in the reporting metadata. Knowing the attributes and measures that correspond to items of a certain report, it is possible to determine the appropriate dimensions and fact tables respectively. Hierarchies (from zero to many) are related to a dimension. Data warehouse schema is defined through association with a fact table or dimension.

It is also possible to determine aggregate functions applied to measures to calculate report items. These aggregate functions are derived from the attribute CalculationFormula of the class Item in the reporting metadata.

### 5.2.4. 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 to move along the hierarchies. Besides, it is desirable that users can modify or construct reports themselves from elements, which are familiar to them, this way, making the report creation more 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.

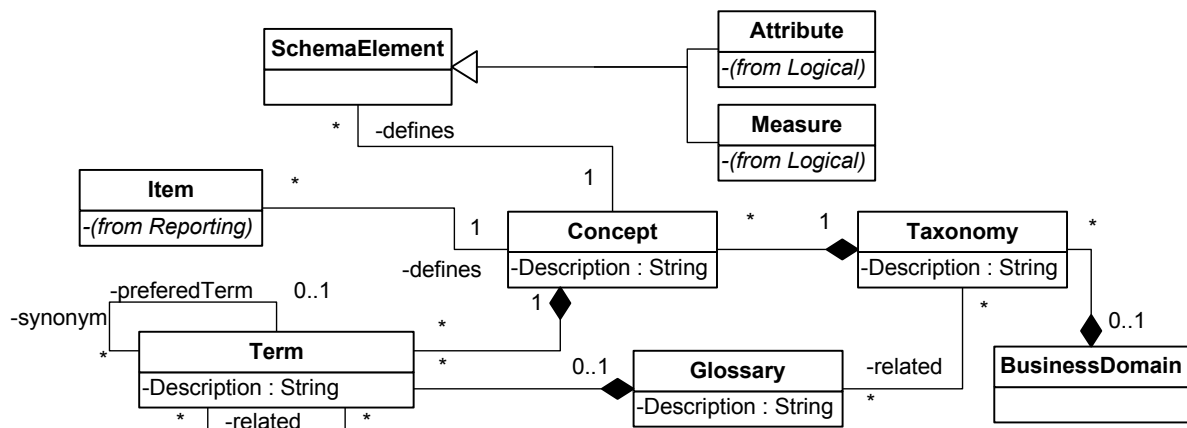


Fig. 5.2.4.1. Semantic metadata [KS12]

Semantic metadata stores the description of the data warehouse elements in business language. In CWM there is the Business Nomenclature package, which can be used to represent business metadata. This package was taken as a basis for semantic metadata depicted in Figure 5.2.4.1. The main classes that are employed to describe 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 (classes Attribute and Measure from the logical metadata) and items used in reports (class Item from the reporting metadata).

### 5.2.5. OLAP Preferences Metadata

A metamodel that describes OLAP preferences is depicted in Figure 5.2.5.1. In this section a revised version of the metamodel [KS12] is presented.

A user may set the degree of interest (DegreeOfInterest, *DOI*) defined in [KI04] 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. For instance, a user operates with values of the DOI attribute that may be the following: very low, low, medium, high, and 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.

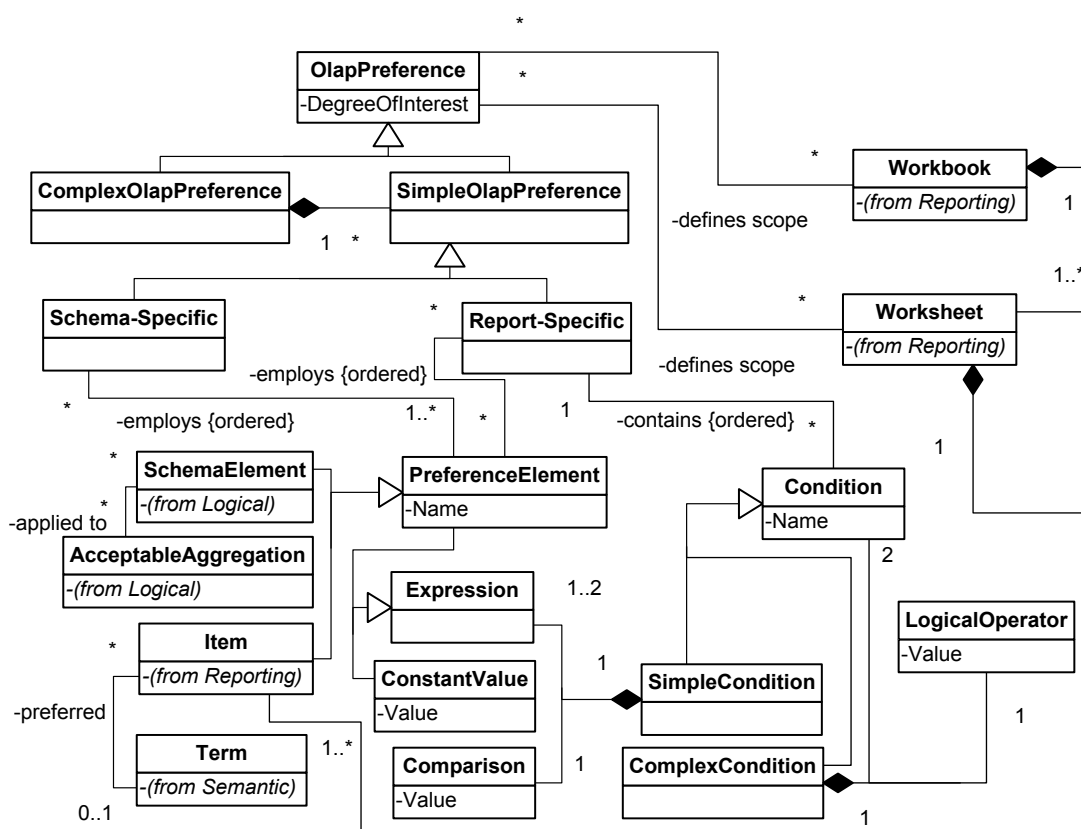


Fig. 5.2.5.1. OLAP preferences metadata [KS12]

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 7), whereas *year=2014* 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.), a particular aggregate function 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. OLAP schema elements correspond to report items (see Figure 5.2.2.1, and section "Interconnection of Report Items and OLAP Schema Elements" for more details). 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 while setting his/her preferences. One or more preference elements may be included in a preference, and vice versa, a single preference element may be used in multiple user preferences.

As report-specific preferences include restrictions on report 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 (i.e. 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*.

### ***OLAP Preference Examples***

To motivate and illustrate the OLAP preference metamodel, several user preference modeling scenarios demonstrated with preference examples have been drawn up. The scenarios were worked out on the basis of the empirical studies of data warehouse reporting tools Oracle Discoverer and the OLAP reporting tool developed in the University of Latvia. For more clearness, the author suggests to display each example as a table with OLAP preference metamodel elements depicted as follows:

- The simple or complex OLAP preference class is in the rightmost column;
- The subclasses or associated classes of either simple or complex OLAP preference are in all the rest columns, excluding the leftmost one;
- Instances of the most specific classes of the OLAP preferences metamodel are in the leftmost column.



The values of the degree of interest (DOI) are omitted in these scenarios, as the principles of assigning and distributing the DOI need a proper explanation, which is proposed further in the thesis (see section 6).

**Scenario A.**

*Description:* A user preference contains solely an OLAP schema element or an aggregate function.

*Type:* schema-specific.

*Scope:* all worksheets in all workbooks.

This is a schema-specific preference that may refer to some element(s) of the OLAP schema or an aggregate function(s) in all available reports.

*Example A.* A user is interested in Program dimension, which contains descriptive attributes of study program. This statement is formulated in Table 6.

**Table 6. A formally described preference from the Example A**

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

**Scenario B.**

*Description:* A user preference 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.

This is a schema-specific preference that may refer to some element(s) of the OLAP schema or an aggregate function(s) in one or multiple sets of reports.

*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 in Table 7.

**Table 7. 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 user preference contains restrictions on data with a simple condition.

*Type:* report-specific.

*Scope:* one or many certain workbooks.

This is a report-specific preference that may refer to one or multiple sets of reports that contain a defined data value of the given report item.

**Table 8. A formally described preference from the Example C**

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>				
Program	Item			Report-Specific	Complex OLAP Preference
Semester	Item	Expression	Simple Condition		
=	Comparison				
'2014-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, and prefers reports that contain study programs. The workbook that contains reports on students' registrations is entitled Registrations. The complex preference set for the Registrations workbook is: *Semester item value is equal to '2014-Spring' by Program*, and apparently it consists of two simple OLAP preferences: (i) *Semester item value*

is equal to '2014-Spring', and (ii) *Study Program should be present in the report*. These statements are formulated in Table 8.

**Scenario D.**

*Description:* A user preference contains restrictions on data with a complex condition.

*Type:* report-specific.

*Scope:* one or many certain worksheets.

This is a report-specific preference that may refer to one or multiple reports that contain a defined data value of the given report item.

**Table 9. A formally described preference from the Example D**

<i>Instance</i>	<i>OLAP Preferences Metamodel Class</i>					
Program	Item			Complex Condition	Report- Specific	Complex OLAP Preference
Faculty	Item					
Program	Item	Expression	Simple Condition			
LIKE	Comparison					
'%Masters%'	Constant Value	Expression	Simple Condition			
AND	Logical Operator					
Year	Item	Expression				
=	Comparison		Simple Condition			
'2013'	Constant Value	Expression				
Statistics	Workbook					
Graduated Students	Worksheet					

*Example D.* Assume that the worksheet entitled Graduated Students of the Statistics workbook reflects yearly data on the total number of students 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' and Year item is set to '2013'*, (ii) *Reports with Faculties included are preferable*, and (iii) *Reports with Study Programs included are preferable*. These statements are formulated in Table 9.

**Scenario E.**

*Description:* A user preference contains restrictions on data with a complex condition.

*Type:* report-specific.

*Scope:* all worksheets in all workbooks.

This is a report-specific preference that may refer to all available reports that contain a defined data value of the given report item.

*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 in Table 10.

**Table 10.** 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						

The above-mentioned scenarios demonstrate the elements that make up either schema-specific or report-specific preferences of varying complexity (i.e. simple/complex) and indicate a scope that contains metadata specified in a preference for further analysis (i.e. one or many certain worksheets/one or many certain workbooks/all worksheets in all workbooks).

### 5.3. Technical Details on the OLAP Reporting Tool

The architecture of the reporting tool is composed of the server with a relational database to store data warehouse data and metadata, data acquisition procedures that 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.

## **5.4. Summary of the Section**

In this section the five different layers of metadata that intersect each other were presented: logical metadata that describes data warehouse schemas, physical metadata that describes storage of a data warehouse in a 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 schemas, and OLAP preferences metadata that stores definitions of user preferences on report structure and data. Various scenarios of formulating OLAP preferences were introduced.

The OLAP preference metamodel is partially used to construct the user preferences in section 6.2. In terms of this thesis, the OLAP preferences that are collected and employed to generate recommendations on reports are simple schema-specific OLAP preferences. 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 [JRTZ09], and report-specific preferences can be constructed according to the metamodel. In its turn, methods for processing schema-specific OLAP preferences described in section 6.2 are the original contribution of this thesis.

## 6. METHODS FOR GENERATION OF RECOMMENDATIONS IN THE OLAP REPORTING TOOL

### 6.1. The Intent of the Section

The intent of this section is to present content-based methods for construction of recommendations for reports in the OLAP reporting tool. Recommendations are generated based on preference information in user profile, which is updated either implicitly or explicitly depending on the method. Taking advantage of data about user preferences for data warehouse schema elements, existing reports that potentially may be interesting to the user are distinguished and recommended. The approach used for recommending reports is composed of three distinct methods – *cold-start*, *hot-start*, and *semantic hot-start* described in detail in terms of this section.

### 6.2. The Proposed Methods for Providing Report Recommendations

Methods presented in the thesis and implemented in the OLAP reporting tool fall into category of the content-based filtering. Users of the reporting tool may have various skill levels (e.g. expert, novice), which is why different methods for generating report recommendations based on user preferences are applied. Methods for providing report recommendations involve implicitly acquired user preferences (i.e. gained automatically from user activity log) that make up a user profile, and methods for stating user preferences explicitly (i.e. provided directly by the user). An evaluation of both types of methods to acquire user preferences (i.e. implicit and explicit) was performed. A detailed description of the experimental study and its results are put forward in section 7.

Each of the methods is exploited in the *mode* in which a user receives recommendations in the reporting tool. However, there are three methods and four modes, because one of the modes employs a combination of two methods. Let's consider each of the modes and their underlying methods for generating report recommendations.

The *user activity mode* employs the *hot-start* method for generation of recommendations. It is applied for a user who has had a rich activity history within the reporting system.

The *report structure mode* employs the *cold-start* method for generation of recommendations. It is applied when (i) a user of the reporting tool starts exploring the system for the first time, or (ii) a user has previously logged into the system, but he/she has been rather passive (the number of activity records is lower than some threshold value). The cold-start method does not exploit user activity history, because in case (i) it is impossible to

generate recommendations by analyzing user previous activity, since it is absent, and in case (ii) poor history of user activity does not reflect user interests in full measure, which may lead to either one-sided or too general recommendations, thereby affecting its quality.

The *automatic mode* is assigned by default to every new user. In automatic mode a user receives recommendations as in report structure mode (exploiting the cold-start method) until crossing a threshold, and then – the user activity mode (exploiting the hot-start method) is employed. A threshold, in fact, is a borderline between the two modes. It is defined as a positive constant, which represents the number of records in the log-table belonging to a certain user, and is considered to be sufficient to switch from one mode to another. Threshold value is a subject to discuss because of various factors that might affect it, e.g. the number of records generated in the log-table while executing a report, the number of available reports according to user rights, the overall number of reports in the reporting tool, the number of users, etc. One should choose a threshold value taking into consideration peculiarities of a particular data warehouse and its reports. The methods reflected in sections 6.2.1. and 6.2.2. are published in [KS11].

In *semantic mode* semantic metadata is considered as a means of formulating user preferences for data warehouse reports explicitly applying a pre-defined description of data warehouse elements. To be more precise, a user formulates his/her preferences employing understandable business 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). Later preferences are processed using an adopted and adjusted algorithm from hot-start method (referred as semantic hot-start method). The description of this approach can be found in [KS12].

### **6.2.1. Hot-Start Method**

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 stored in a log-table and gathered in a user profile. Secondly, reports that are composed of data warehouse schema elements, which are potentially the most interesting to a user, are determined.

In the hot-start method, weights of schema elements are used to propagate the degree of interest from sub-elements to the elements of higher level. When a new schema is defined in the data warehouse repository, weights of the new schema elements are calculated and weights of the existing schema elements are adjusted. A *weight* of a schema element is computed in the following way:

- The weight of a schema  $S_i$  equals to  $W(S_i) = 2$ , since the total weight of all fact tables is 1 and so does the total weight of all dimensions related to schema  $S_i$ .
- The weight of a fact table  $F_i$  equals to  $W(F_i) = \frac{1}{n}$ , where  $n$  is the number of fact tables belonging to one schema.
- Since a dimension can belong to multiple schemas, the weight of a dimension is calculated separately for each schema, which a dimension belongs to. The weight of a dimension  $D_i$  in a schema  $S_j$  equals to  $W(D_i, S_j) = \frac{1}{k \cdot m_i}$ , where  $k = \sum_{l=1}^n \frac{1}{m_l}$ ,  $n$  is the number of dimensions belonging to the schema  $S_j$ , and  $m_i \in m_1, \dots, m_n$  is the number of schemas, to which the dimension  $D_i$  is related. The number of schemas, which a dimension belongs to, is taken into account, because it is assumed that a dimension used in multiple schemas is less specific for the particular schema. For example, time dimension is almost always involved in every schema in a data warehouse and it is not specific for any of the schemas.
- The weight of a measure  $M_i$  of a fact table  $F_j$  equals to  $W(M_i, F_j) = \frac{1}{n}$ , where  $n$  is the number of measures belonging to the fact table  $F_j$ .
- The weight of an attribute  $A_i$  of a dimension  $D_j$  equals to  $W(A_i, D_j) = \frac{1}{n}$ , where  $n$  is the number of attributes belonging to the dimension  $D_j$ .
- To compute the degree of interest of a hierarchy, the weight of each attribute in that hierarchy is used. The weight of an attribute  $A_i$ , which is a level of a hierarchy  $H_j$ , equals to  $W(A_i, H_j) = \frac{W(A_i, D_k)}{n}$ , where  $n$  is the number of attributes that make up levels of the hierarchy  $H_j$ , and  $D_k$  is the dimension, which the attribute  $A_i$  belongs to. Basically, the weight of an attribute in a hierarchy is the weight of the attribute in a dimension divided by the number of levels in the hierarchy.



### **Discovering User Preferences**

The degree of interest in OLAP user preferences by analyzing user behaviour in the reporting system is maintained and updated. When a user runs a report, items of the report are obtained by means of the reporting metadata analysis. After schema elements used in the report are determined as described in section “Interconnection of Report Items and OLAP Schema Elements”, user’s degree of interest for each schema element employed in the report is updated hierarchically, starting from the elements of the finer levels of granularity. An update of the degrees of interest is conducted according to the Algorithm 1, which is executed for each attribute or measure used in the report.

#### *Algorithm 1*

*Input:* User OLAP preferences for schema elements with the degrees of interest for each element and the schema element  $E$  used in a report that was executed by the user.  $DOI(SE)$  is the user’s degree of interest for the schema element  $SE$  according to the user profile. In case of dimensions:  $DOI(SE, S)$ , where  $SE$  is a dimension and  $S$  is a particular schema, which this dimension refers to.

*Output:* User OLAP preferences with updated degrees of interest.

```
// if element E is a measure
if E instanceof(Measure) then
    DOI(E)=DOI(E)+1;
    // getting a fact table, which the measure E belongs to
    F=getFactTable(E);
    DOI(F)=DOI(F)+W(E);
    // getting a schema, which the fact table F belongs to
    S=getSchema(F);
    DOI(S)=DOI(S)+W(F)*W(E);
// 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(E,D)=DOI(E,D)+1;
    // getting a schema, which the dimension D belongs to
    S=getSchema(D);
    DOI(D,S)=DOI(D,S)+W(E,D);
    DOI(S)=DOI(S)+W(D,S)*W(E,D);
```

```
// getting hierarchies, levels of which correspond to the attribute E
hierarchies=getHierarchies(E);
foreach H in hierarchies do
    DOI(H)=DOI(H)+W(E,D)/countLevels(H);
end loop;
end if;
```

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 an acceptable aggregation is obtained in the logical metadata, and its degree of interest is increased by 1.

### ***Recommending Reports***

When degrees of interest are updated in the user's OLAP preferences, the user profile is compared with all reports defined in the reporting metadata and reports, which are potentially interesting for the user, are determined.

The content-based filtering approach [VM03] is widely used in item-based recommender systems to classify the items into potentially interesting/uninteresting. Data warehouse specifics is cardinally different from that of the recommender systems, for that reason, the hot-start method is an original method that applies the principles of the content-based filtering approach in the context of a data warehouse.

User's schema-specific OLAP preferences are compared with schema elements used in each report to estimate the *hierarchical similarity* between a user profile and a report. The hierarchical similarity between a report and a user profile depends on the number of schema elements used in the report and the degree of interest for these elements set in user profile. Data warehouse schema elements that were used in the report are determined similarly as described in the section "Interconnection of Report Items and OLAP Schema Elements".

The algorithm for comparing reports with user profiles is based on the following assumptions:

- If the user's degree of interest for a measure  $M$  is 0, but the degree of interest for a fact table  $F$  containing  $M$  is positive, then the user might be also interested in  $M$ .
- If the user's degree of interest for an attribute  $A$  is 0, but the degree of interest for a dimension  $D$  containing  $A$  is positive, then the user might be also interested in  $A$ .
- If the user's degree of interest for an attribute  $A$  is 0, but the degree of interest for a hierarchy  $H$  containing  $A$  is positive, then the user might be also interested in  $A$ .

- If the user's degree of interest for a dimension or fact table  $E$  is 0, but the degree of interest for a schema  $S$  containing  $E$  is positive, then the user might be also interested in  $E$ .

To calculate the hierarchical similarity, the formula employed to compute the user-item similarity score for items defined by a hierarchical ontology [MSST10] was considered in the context of the OLAP schema elements. In [MSST10] authors deal with hierarchical ontology of news, they collect concepts (for instance, "life style", "politics", "crime", "elections", etc.) that a user is interested in into user profile and compare them to concepts in items, i.e. newspapers. In terms of this method, 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"). Thus, the hierarchical similarity between a report and a user profile in terms of this thesis is computed as follows (Formula 1):

$$sim = \frac{\sum_{i=1}^n DOI(E_i)}{\sum_{j=1}^m DOI(G_j)}, \quad (1)$$

where  $E_1, \dots, E_n$  are schema elements used in the report, and  $G_1, \dots, G_m$  are all schema elements in the user profile.

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). However, the essence of any data warehouse is in facts, while the describing attributes give the auxiliary information. Thereby, it is assumed that the *TopN* recommendations can be filtered (i) firstly, by the value of the fact-based similarity coefficient, (ii) secondly, by the one of dimension-based similarity coefficient, and (iii) finally, by aggregate function DOI.

To demonstrate the 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 6.2.1.1.) 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 enrolment actions and the number of enrolment cancellations for each course (dimension *Course*), study program (dimension *Program*), status (dimension *Status*), and date (dimension *Time*).

Dimensions *Time* and *Program* contain hierarchies with corresponding levels, which are shown in Figure 6.2.1.2.

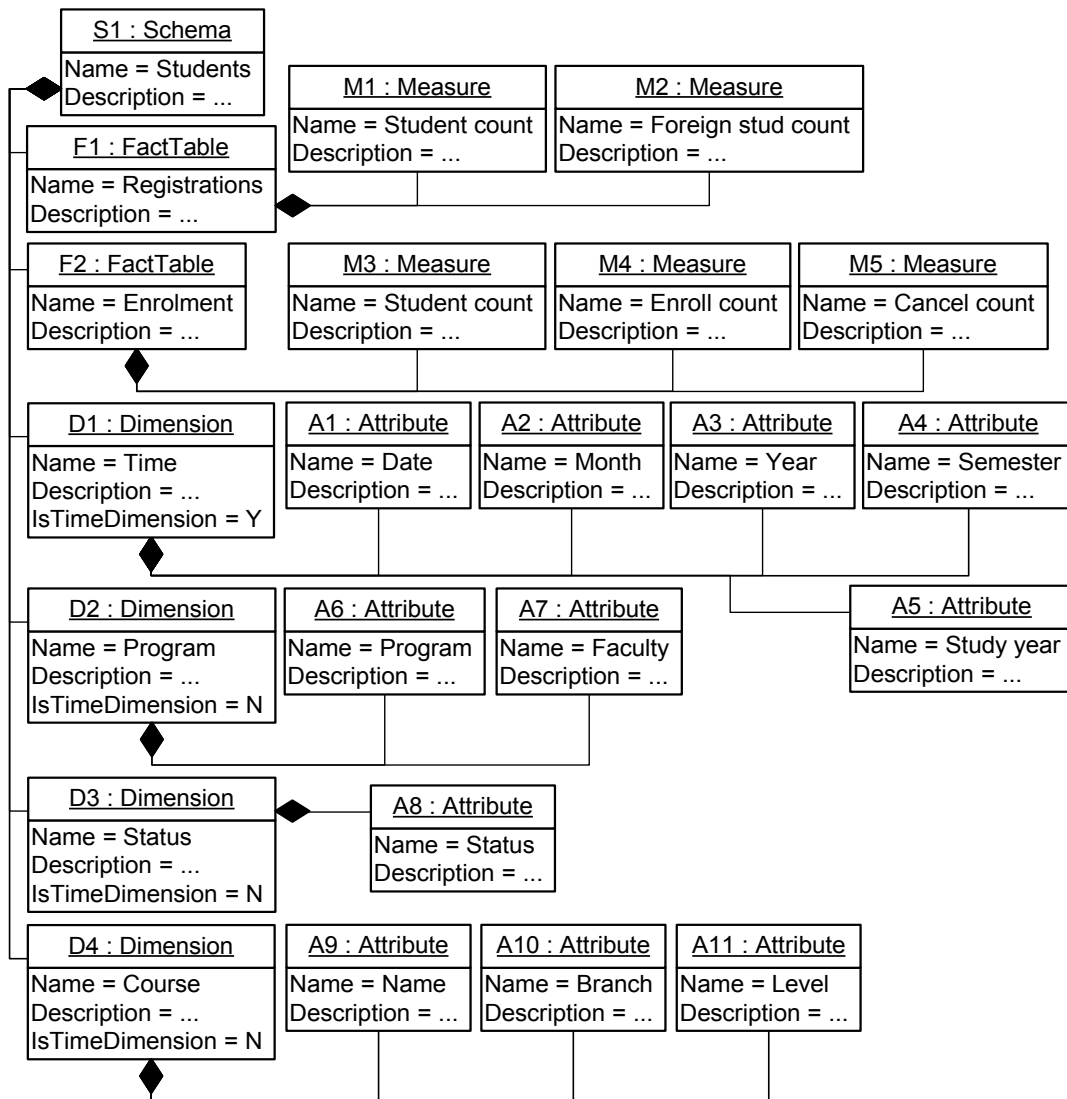


Fig. 6.2.1.1. *Students* data warehouse schema

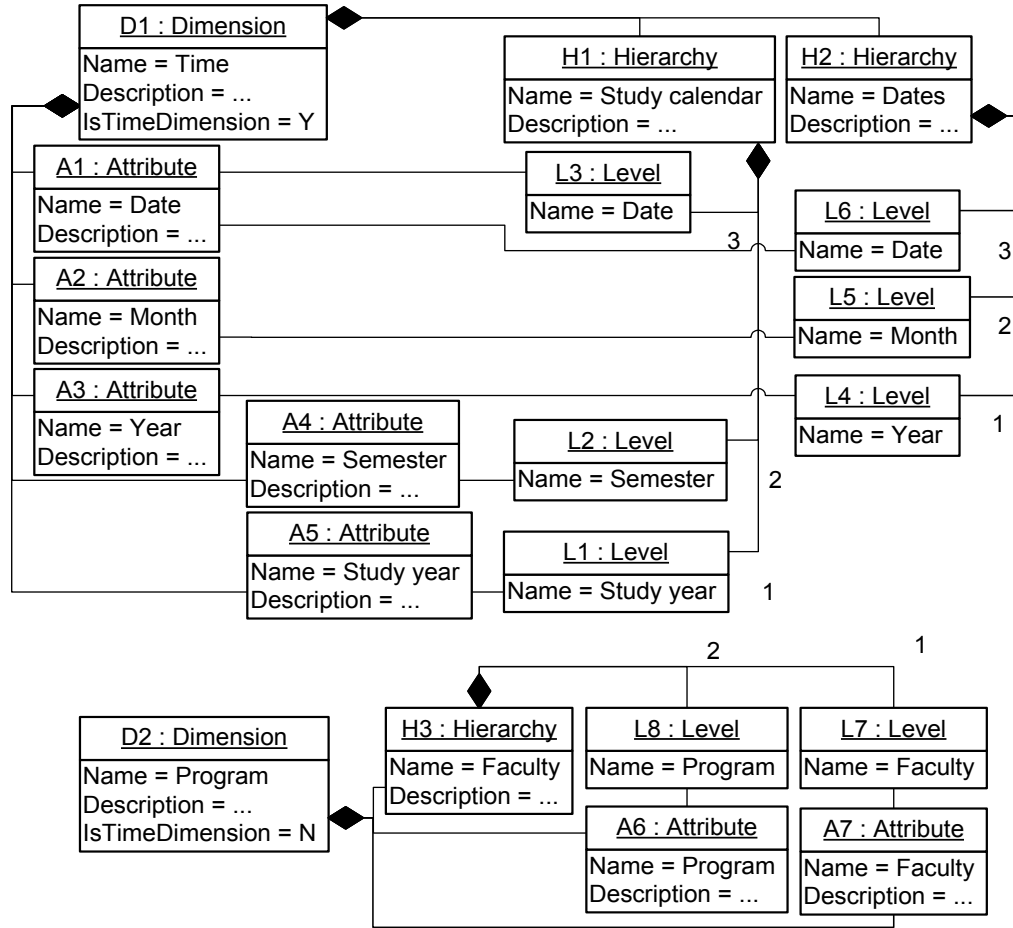


Fig. 6.2.1.2. Hierarchies of the dimensions *Time* and *Program*

Let’s now compute weights of the schema elements. Suppose that *Time* dimension is used in 3 other schemas additionally to the *Students* schema, and other dimensions *Program*, *Status*, and *Course* belong only to the *Students* schema. Weights of the elements are shown in Table 11 and weights of the hierarchy levels are shown in Table 12.

Table 11. Weights and DOI of Students data warehouse schema and its elements

	Schema	Fact tables		Measures					Dimensions				Attributes										
	S <sub>1</sub>	F <sub>1</sub>	F <sub>2</sub>	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	A <sub>10</sub>	A <sub>11</sub>
<b>Weight</b>	2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
<b>DOI</b>	$\frac{4723}{780}$	$\frac{7}{2}$	3	0	7	5	0	4	$\frac{9}{5}$	2	5	$\frac{5}{3}$	0	1	4	4	0	0	4	5	4	1	0

Assume that user’s degrees of interest for the schema elements computed by the algorithm 1 are such as shown in Table 11 row DOI, and the user’s degrees of interest for hierarchies with levels composed of attributes are such as shown in Table 12 row DOI.

**Table 12. Weights of hierarchy levels and DOI of the hierarchies**

	<i>Hierarchies</i>			<i>Attributes/Hierarchy Levels</i>							
				<i>Hierarchy H<sub>1</sub></i>			<i>Hierarchy H<sub>2</sub></i>			<i>Hierarchy H<sub>3</sub></i>	
	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>1</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>7</sub>	A <sub>6</sub>
<i>Weight</i>				$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{4}$	$\frac{1}{4}$
<i>DOI</i>	$\frac{4}{15}$	$\frac{1}{3}$	1								

To calculate the hierarchical similarity, let's consider two example reports. The first report is  $R_1$  – Average foreign student count for each study program per semester. The second report is  $R_2$  – Total student count enrolled into courses for each faculty per year. The hierarchical similarity values for the reports  $R_1$  and  $R_2$  are computed separately for fact-based recommendations  $simF_{R_1}$  (Formula 2) and  $simF_{R_2}$  (Formula 3), and for dimension-based recommendations  $simD_{R_1}$  (Formula 4) and  $simD_{R_2}$  (Formula 5) respectively.

$$simF_{R_1} = \frac{DOI(M_2) + DOI(F_1) + DOI(S_1)}{DOI(S_1) + DOI(F_1) + DOI(F_2) + DOI(M_1) + \dots + DOI(H_3)} \approx 0.26 \quad (2)$$

$$simF_{R_2} = \frac{DOI(M_3) + DOI(F_2) + DOI(S_1)}{DOI(S_1) + DOI(F_1) + DOI(F_2) + DOI(M_1) + \dots + DOI(H_3)} \approx 0.22 \quad (3)$$

$$simD_{R_1} = \frac{DOI(S_1) + DOI(D_2) + DOI(A_6) + DOI(H_3) + DOI(D_1) + DOI(A_4) + DOI(H_1)}{DOI(S_1) + DOI(F_1) + DOI(F_2) + DOI(M_1) + \dots + DOI(H_3)} \approx 0.24 \quad (4)$$

$$simD_{R_2} = \frac{DOI(S_1) + DOI(D_2) + DOI(A_7) + DOI(H_3) + DOI(D_1) + DOI(A_3) + DOI(H_2)}{DOI(S_1) + DOI(F_1) + DOI(F_2) + DOI(M_1) + \dots + DOI(H_3)} \approx 0.30 \quad (5)$$

According to the fact-based similarity values between OLAP preferences in the user profile and reports  $R_1$  and  $R_2$ , the report  $R_1$  is ranked higher than the report  $R_2$ , but in compliance with the dimension-based similarity values, these reports are ordered the other way. Thus, one should take into consideration the order of the similarity values.

### 6.2.2. Cold-Start Method

Instead of parsing user activity as in the hot-start method, the cold-start method is proposed, which is suitable for the user who is either new or a passive one (i.e. a user whose

number of activity records is lower than some pre-defined threshold value). The essence of cold-start method is composed of two components: firstly, structural analysis of existing reports is performed, and secondly, likeliness between each pair reports is revealed.

The cold-start method addresses two issues most common in recommender systems: a *new item* (or *long-tail* as in [PT08]) issue and a *cold-start* user (i.e. a user with no previous activity in the system) issue. The main point of a new item or long-tail issue in recommender systems is that items, which are either newly added to the system or unpopular (i.e. received too few rating set by users), are practically of no use, because the overall rating score based on user ratings is either absent or too low. As a result, the number of items that are never recommended (a *long tail*) to users increases. In the cold-start method described in this section the new item issue along with the cold-start user issue is solved, since the likeliness between reports is defined irrespective of user activity. More precisely, similarity scores that reflect likeliness are recalculated each time a new report is being created, an existing report is being deleted or any kind of changes in existing reports are being made.

In the cold-start method, *report structure* denotes data warehouse schema elements and acceptable aggregate functions, which are related to items of a certain report. OLAP schema elements used in a report are discovered as described in section “Interconnection of Report Items and OLAP Schema Elements”, and report structure is defined. Each report is represented as a *Report Structure Vector (RSV)* by Formula 6, which is of the following form:

$$RSV = (e_{11}, e_{12}, \dots, e_{1k_1}, \dots, e_{n1}, e_{n2}, \dots, e_{nk_n}), \quad (6)$$

where  $e_{ik_i}$  is a vector coordinate, i.e. a binary value that indicates presence (equals 1) or absence (equals 0) of the instance of the report structure element,  $k_i$  is the number of elements in  $i$ -th structure,  $i$  is the index number of each structure ( $i = 1, 2, \dots, n$ ),  $n$  is the total number of distinct structure elements in reports. In a typical case,  $n = 7$  as there is a finite set  $S$  of 7 elements,  $S = \{\text{attribute, measure, fact table, dimension, schema, acceptable aggregation, hierarchy}\}$ .

Two instances of *RSV* depicted in Figure 6.2.2.1 provide an example of *RSV* application:

- Vector  $\bar{r}_1$  describes the structure of the report *R1 – Average student count for each faculty per semester*,
- Vector  $\bar{r}_2$  describes the structure of the report *R2 – Total PhD student count for each study program per year*.





Similarity value,  $sim$ , of the pair of vectors  $\vec{r}_1$  and  $\vec{r}_2$  in Figure 6.2.2.1 is:

$$sim = \frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1| * |\vec{r}_2|} = \frac{8}{\sqrt{11} * \sqrt{11}} \approx 0,727 .$$

### **Discovering Similarities**

The cold-start approach is oriented on providing users with recommendations while working with a certain report. Assume that the report browsed by the user at the moment is called an *active* report. Thus, in order to generate cold-start recommendations, similarity is calculated by means of report structure vectors (*RSV*) among the active report and all the rest of the data warehouse reports that the user has a right to access. Taking into consideration the facts that (i) a new report might be created, (ii) some of the existing reports may get deleted, (iii) there might be changes in existing report structure, *RSV* and *sim* values have to be recalculated dynamically every time any of the mentioned events takes place.

### **Recommending Reports**

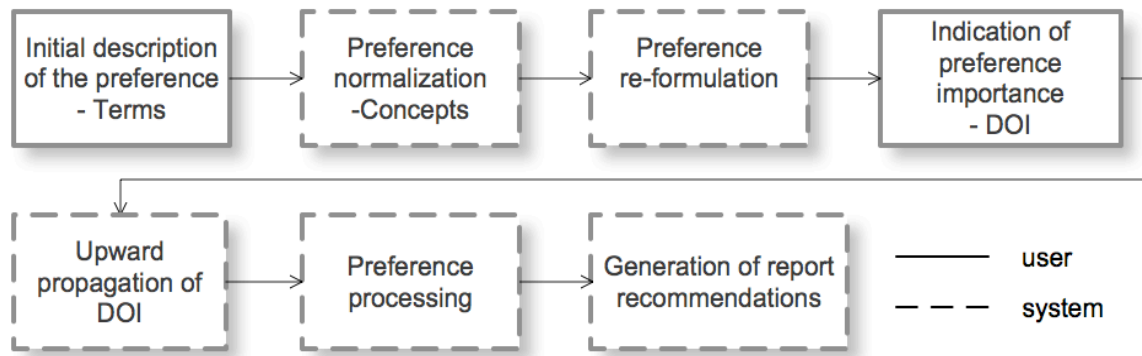
Finally, a list of *TopN* report recommendations with  $N$  highest *sim* values sorted in descending order is returned to the user. Note that if a group of users have similar rights on reports, then for each of the users the recommendation list will be the same, as, in case of the cold-start method, solely the structure of reports as such has an impact on the recommendations.

### **6.2.3. Semantic Hot-Start Method**

In the reporting tool 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 section 5.2.4.).

#### **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.



**Fig. 6.2.3.1.** Processing user preferences stated with semantic metadata

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 [GBR07] and altered. In [GBR07] authors 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 [GBR07] 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 thesis 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*. Later preferences are processed using an adopted and adjusted algorithm from hot-start method. The process of preference creation and transformation is depicted in Figure 6.2.3.1. and is an improved version from that in [SK11].

***Processing of User Preferences Stated with Semantic Metadata by Means of Semantic Hot-Start Method***

The process of explicit preference creation and transformation (see Figure 6.2.3.1.) is explained in this section. For explicitly defined schema-specific preferences, it is possible to apply the adapted hot-start method for providing recommendations on reports, which is based on explicitly stated preferences in the user profile instead of implicitly discovered schema-specific preferences described in [KS11] and in section 6.2.1. A quick reference to the hot-

start method – it 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, reports that are composed of data warehouse schema elements, which are potentially the most interesting to a user, are determined.

Let’s refer to an adapted hot-start method (see section 6.2.3.) for explicitly defined user preferences as a semantic hot-start method. The main differences between the hot-start and semantic hot-start methods are as follows. In case of semantic hot-start method, the first step of the hot-start method is not applicable and must be substituted since users specify preferences themselves. In the first step, the semantic hot-start method should process user preferences for schema elements of a finer level of granularity and propagate degrees of interest 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 the level. This propagation should be proportional to the number of levels in the hierarchy. More details on DOI propagation are available in *Step 5: Upward Propagation of DOI*. The second step of the semantic 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. See *Step 6: Preference Processing* for more details.

### ***Steps for Processing User Preferences Described with Semantic Metadata***

This section gives a consequent description of all the steps (see Figure 6.2.3.1.) that should be performed to process user preferences defined with semantic data.

**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 5.2.5.1.). To limit the set of terms that are proposed for a user to formulate preferences, the user should select a glossary that contains terms and seems to be the most suitable and understandable for him/her (see Figure 5.2.4.1.). Next, a user describes his/her preference choosing one of the synonym terms from the glossary.

*Example:* terms “study program”, “academic specialization”, “branch”, “field of study” are considered synonyms, from which a user is free to select the most appropriate one.

**Step 2: Preference Normalization.** A set of terms corresponds to exactly one concept (see Figure 5.2.4.1.). 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 5.2.4.1.) 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 5.2.5.1, 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, thus, providing a typical numerical scale for assessment of the DOI. Quantitative values of the DOI are employed for further processing of preferences.

**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 (see section 5.2.3.) 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 2 that provides upward propagation of the DOI 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 2*

*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  $F_i$  equals to

$$DOI(F_i) = \sum_{j=1}^k \frac{DOI(M_j)}{n}, \text{ where } DOI(M_j) \text{ are the values of the DOI of measures belonging to}$$

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  $D_i$  equals to

$$DOI(D_i) = \sum_{j=1}^k \frac{DOI(A_j)}{n}, \text{ where } DOI(A_j) \text{ are the values of the DOI of attributes belonging to a}$$

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  $H_i$  equals to

$$DOI(H_i) = \sum_{j=1}^k \frac{DOI(A_j, D_l)}{n}, \text{ where } D_l \text{ is the dimension, which the attribute } A_j \text{ belongs to,}$$

$DOI(A_j, D_l)$  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  $S_i$  equals to

$$DOI(S_i) = \sum_{j=1}^k \frac{DOI(D_j)}{d} + \sum_{l=1}^m \frac{DOI(F_l)}{f}, \text{ where } DOI(D_j) \text{ are the values of DOI of dimensions}$$

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  $S_i$ .

A user may state in the profile the DOI of 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 6.2.1.2) 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 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. To calculate the similarity score between a report and a user profile, the hierarchical similarity (see section 6.2.1.) between a report and a user profile is computed as shown in Formula 8:

$$sim = \frac{\sum_{i=1}^n DOI(E_i)}{\sum_{j=1}^m DOI(P_j)}, \quad (8)$$

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

**Step 7: Generation of Report Recommendations.** Similar to recommendations produced by means of hot-start method, it is assumed that the *TopN* recommendations can be filtered (i) firstly, by the value of the fact-based similarity coefficient, (ii) secondly, by the one of dimension-based similarity coefficient, and (iii) finally, by summarized DOI for aggregate functions applied to the measures of the report.

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 13.

Preferences P1-P8 are mapped to elements of *Students* schema, whereas P9 refers to *Gradebook* schema. The calculation of hierarchical similarity values by means of semantic hot-start method will be illustrated on two reports that were presented in section 6.2.1:  $R_1$  – *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.

**Table 13. Mapping of semantic metadata elements and data warehouse schema elements**

	<i>Semantic Metadata Element</i>		<i>Schema Element</i>		<i>Aggregate Function</i>	<i>DOI</i>
	<i>Term</i>	<i>Concept</i>	<i>Attribute</i>	<i>Measure</i>		
<b>P1</b>	Academic specialization	Study program	Program	-	-	0.75 (high)
<b>P2</b>	Faculty	Faculty	Faculty	-	-	0.9 (very high)
<b>P3</b>	Number of students	Student count	-	Student count	-	0.5 (average)
<b>P4</b>	Year	Year	Year	-	-	1 (very high)
<b>P5</b>	Number of foreign students	Foreign student count	-	Foreign stud count	-	0.4 (low)
<b>P6</b>	-	-	-	-	SUM	0.85 (very high)
<b>P7</b>	-	-	-	-	AVG	0.35 (low)
<b>P8</b>	Course title	Course	Name	-	-	0.55 (average)
<b>P9</b>	Average student grade	Average student grade	-	Average stud grade	-	0.7 (high)

**Table 14. DOI values of Students data warehouse schema and its elements**

	<i>Schema</i>	<i>Fact Tables</i>		<i>Measures</i>					<i>Dimensions</i>				<i>Attributes</i>										
	S <sub>1</sub>	F <sub>1</sub>	F <sub>2</sub>	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	A <sub>10</sub>	A <sub>11</sub>
<b>DOI</b>	$\frac{233}{480}$	$\frac{2}{10}$	$\frac{1}{6}$	0	$\frac{4}{10}$	$\frac{1}{2}$	0	0	$\frac{1}{5}$	$\frac{33}{40}$	0	$\frac{11}{60}$	0	0	1	0	0	$\frac{3}{4}$	$\frac{9}{10}$	0	$\frac{11}{20}$	0	0



Employing an Algorithm 2 for propagation of DOI from the elements of the finer level of granularity (attributes and measures; see Table 13), the DOI values for the elements of the coarser level of granularity (dimensions, fact tables, hierarchies, schema; see Figure 6.2.1.1. and Figure 6.2.1.2. in section 6.2.1.) are computed. In Table 14 the values of user's degree of interest (DOI) for all attributes, dimensions, fact tables, and a schema itself are shown.

The values of user's degree of interest (DOI) for hierarchies with levels composed of attributes are such as shown in Table 15.

**Table 15. DOI values of hierarchy levels and hierarchies of Students data warehouse schema**

	Hierarchies			Attributes/Hierarchy Levels							
				Hierarchy H <sub>1</sub>			Hierarchy H <sub>2</sub>			Hierarchy H <sub>3</sub>	
	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>1</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>7</sub>	A <sub>6</sub>
<b>DOI</b>	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 9) and  $simF_{R2}$  (Formula 10), and for dimension-based recommendations  $simD_{R1}$  (Formula 11) and  $simD_{R2}$  (Formula 12) respectively.

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

$$simF_{R1} = \frac{DOI(M_2) + DOI(F_1) + DOI(S_1)}{DOI(p)} \approx \frac{1.085}{6.31875} \approx 0.17 \tag{9}$$

$$simF_{R2} = \frac{DOI(M_3) + DOI(F_2) + DOI(S_1)}{DOI(p)} \approx \frac{1.152}{6.31875} \approx 0.18 \tag{10}$$

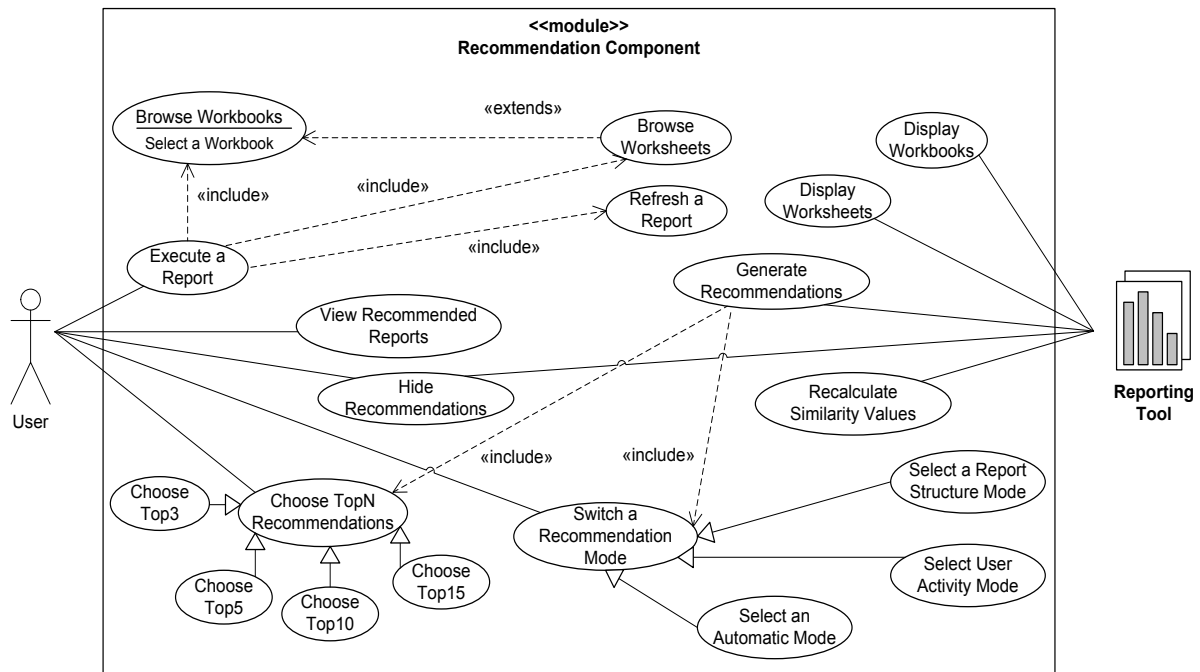
$$simD_{R1} = \frac{DOI(S_1) + DOI(D_2) + DOI(A_6) + DOI(H_3) + DOI(D_1) + DOI(A_4) + DOI(H_1)}{DOI(p)} \approx \frac{3.085}{6.31875} \approx 0.49 \tag{11}$$

$$simD_{R_2} = \frac{DOI(S_1) + DOI(D_2) + DOI(A_7) + DOI(H_3) + DOI(D_1) + DOI(A_3) + DOI(H_2)}{DOI(p)} \approx \frac{4.568}{6.31875} \approx 0.72 \quad (12)$$

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  $R_1$ .

### 6.2.4. Adding a Recommendation Component

To describe user interaction with the recommendation component of the reporting tool, the main actions of both the user and the reporting tool are depicted in Figure 6.2.4.1.



**Fig. 6.2.4.1.** An UML Use Case diagram of the recommendation component of the data warehouse reporting tool

When a user signs in the reporting tool, a set of all workbooks that are accessible for this user in accordance with the access rights are at user's disposal (*Display Workbooks*, *Display Worksheets*). A user may select any workbook (*Browse Workbook*) from the list and browse its worksheets (*Browse Worksheets*) each of which displays a single report. Once the report is executed (*Execute a Report*) or refreshed (*Refresh a Report*), a recommendation component returns to a user several generated recommendations (*Generate Recommendations*, *View Recommended Reports*) for other reports that have some common

OLAP schema elements with the executed one. All recommendations indeed are links to other worksheets formed as *WorkbookName.WorksheetName* followed by a similarity coefficient, and are sorted in a decreasing order of its value.

Executing (or refreshing) the recommended report, a user receives another set of recommendations (*Generate Recommendations, View Recommended Reports*), and so on. The maximum number of recommendations (*Choose TopN Recommendations*) by default is 3 (*Choose Top3*), but the user may adjust it to his/her taste to 5 (*ChooseTop5*), 10 (*Choose Top10*), or 15 (*Choose Top15*). If the user is convinced that recommendations are not needed at the moment, then he/she can turn this option off (*Hide Recommendations*). All recommendation mode settings are being saved and retrieved next time when the user logs into the system.

Due to the fact that (i) a new report might be created or any of the existing reports may be deleted, (ii) there might be changes in existing reports' structure, or (iii) user's activity during the current session or preceding sessions should be analyzed, values of all similarity coefficients have to be recalculated (*Recalculate Similarity Values*). It is implemented as a maintenance procedure is launched dynamically each time when a user signs in or switches to Activity mode, or when some changes take place. User activity data to be analyzed is gathered for the last 12 months to keep recommendations up-to-date, as 1 year is a typical reporting period.

### **6.2.5. Examples of Generated Recommendations**

#### ***Examples of Recommendations in User Activity Mode***

The hot-start method for generation of recommendations is employed in *user activity mode*. It is applied for users who have had a rich activity history (i.e. the number of records in user activity history exceeds the pre-defined threshold value) with the reporting system. As mentioned earlier, a threshold value is a subject to discuss because of various factors that might affect it, such as, for instance, the number of records generated in the log-table while executing a report, the number of available reports according to user rights, and so on. Thus, the pre-defined threshold value serves to distinguish *passive* users from *active* ones, and is exploited in the *automatic mode* for switching between the *report structure* and *user activity modes*.

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 dimension-based similarity coefficient, (ii) secondly, by the value of the fact-based similarity coefficient, and (iii) finally, by aggregate function DOI.

An example of recommendations generated for one of the users in user activity mode is presented in Figure 6.2.5.1. The usage scenario includes 10 recommendations sorted in descending order, first, by the dimension-based similarity coefficient value, then, by the fact-based similarity coefficient, and finally, by aggregate function DOI.

In the example given in Figure 6.2.5.1 values of both dimension-based and fact-based similarity coefficients are relatively low, which signifies that the user didn't have any strong priorities over the reports and was interested in a set of reports belonging to different schemas. The top reports are #1: *Aktīvo lietotāju vidējā aktivitāte pa kursu kategorijām – Average activity of active users by course categories* and #2: *Kopējais uzdevumu skaits mēnesī pa kursiem – Total monthly students' task count by course*. In Moodle CMS context, an *active user* is the one who has logged into the system at least once in the defined period of time.

#### Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji

Aktīvo lietotāju kopējā aktivitāte pa kursu kategorijām	Aktīvo lietotāju vidējā aktivitāte pa kursu kategorijām	Aktīvo lietotāju kopējā aktivitāte pa programmām	Aktīvo lietotāju vidējā aktivitāte pa programmām
---	---	--	--

Page < 1 / 1 > 1 Show Hide Report Recommendations

Nr.	Recommendation by User Activity	Similarity
1	<a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa kursu kategorijām</a>	0.375; 0.338
2	<a href="#">Vērtējumu grāmata - Uzdevumu skaits. Kopējais uzdevumu skaits mēnesī pa kursiem</a>	0.366; 0.181
3	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.341; 0.173
4	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.341; 0.173
5	<a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa programmām</a>	0.326; 0.338
6	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173
7	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173
8	<a href="#">Vērtējumu grāmata - Vērtējumu skaits. Kopējais vērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173
9	<a href="#">Vērtējumu grāmata - Vērtējumu tipi. Vērtējumu tipu sadalījums mēnesī pa kursiem</a>	0.326; 0.173
10	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Gala vērtējumu vērtības pa kursiem ārzemniekiem</a>	0.235; 0.357

**Fig. 6.2.5.1.** An example for recommendations in user activity mode

As dimension-based and fact-based similarity coefficient values may highly differ, they are both shown to the user, for instance, to make him/her aware of the higher extent of

fact-based similarity even if the dimension-based similarity is lower (e.g. reports #5: *Aktīvo lietotāju vidējā aktivitāte pa programmām – Average activity of active users by study program* and #10: *Gala vērtējumu vērtības pa kursiem ārzemniekiem – Final grade values of foreign students by course*).

In its turn, aggregate function DOI coefficient is hidden from the user as it is considered to be less informative but helpful in sorting in case when two or more reports have the same fact-based and dimension-based similarity coefficient values, e.g. reports #6–#9 (#6: *Gala vērtējumu skaits mēnesī pa kursiem – Total monthly students' final grade count by course*, #7: *Starpvērtējumu skaits mēnesī pa kursiem – Total monthly students' interim grade count by course*, #8: *Kopējais vērtējumu skaits mēnesī pa kursiem – Total monthly students' grade count by course*, and #9: *Vērtējumu tipu sadalījums mēnesī pa kursiem – Monthly distribution of students' grade types by course*) have equal dimension-based and fact-based similarity values (respectively, 0.326; 0.173), as do reports #3: *Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem – Total monthly students' final grade count of foreign students by course* and #4: *Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem – Total monthly students' interim grade count of foreign students by course* (respectively, 0.341; 0.173). Such coefficient values illustrate that these groups of reports (#3-#4 and #6-#9) consist of logical metadata with similar total DOI value, whereas restrictions on data in these reports vary.

### **Examples of Recommendations in Report Structure Mode**

In *report structure mode* the cold-start method for generation of recommendations is employed. It is applied when (i) a user of the reporting tool starts exploring the system for the first time, or (ii) a user has previously logged in the system but he/she has been rather passive (i.e. the number of activity records is lower than some threshold value). The usage scenario includes 10 recommendations generated in the report structure mode for one of the reports – *Final grade values by course* (i.e. *Gala vērtējumu vērtības pa kursiem*) – which are depicted in Figure 6.2.5.2. Recommendations are sorted by the similarity coefficient value in descending order.

Note that reports #1: *Starpvērtējumu vērtības pa kursiem – Interim grade values by course* has the similarity coefficient value equal to 1, which in its turn means that the structure of this report is the same (i.e. the same OLAP schema elements are employed). However, in case of high value of similarity coefficient the data still may differ because of various restrictions on data in each of these reports. Also, if synonymic terms that denote the semantic meaning of one and the same OLAP schema element are different, it will not affect the result

(i.e. reports containing the same OLAP schema elements will still have the similarity coefficient value equal to 1).

**Vērtējumu grāmata - Gala un starpvērtējumu vērtības**

<b>Gala vērtējumu vērtības pa kursiem</b>	Starpvērtējumu vērtības pa kursiem	Gala vērtējumu vērtības pa kursiem ārzemniekiem	Starpvērtējumu vērtības pa kursiem ārzemniekiem
---	------------------------------------	---	---

Page  / 1

Nr.	Recommendation by Report Structure	Similarity
1	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Starpvērtējumu vērtības pa kursiem</a>	1.000
2	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Starpvērtējumu vērtības pa kursiem ārzemniekiem</a>	0.926
3	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Gala vērtējumu vērtības pa kursiem ārzemniekiem</a>	0.926
4	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem</a>	0.696
5	<a href="#">Vērtējumu grāmata - Vērtējumu skaits. Kopējais vērtējumu skaits mēnesī pa kursiem</a>	0.696
6	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem</a>	0.696
7	<a href="#">Vērtējumu grāmata - Vērtējumu tipi. Vērtējumu tipu sadalījums mēnesī pa kursiem</a>	0.667
8	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.641
9	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.641
10	<a href="#">Vērtējumu grāmata - Uzdevumu skaits. Kopējais uzdevumu skaits mēnesī pa kursiem</a>	0.583

**Fig. 6.2.5.2.** An example of recommendations in report structure mode (report *Gala vērtējumu vērtības pa kursiem – Final grade values by course*)

The extent of similarity of each report in the Top10 list and the one browsed by user at the moment varies from high (1.000) to medium (0.583). The higher the value of similarity coefficient is (as in #1: *Starpvērtējumu vērtības pa kursiem – Interim grade values by course*, #2: *Starpvērtējumu vērtības pa kursiem ārzemniekiem – Interim grade values of foreign students by course*, and #3: *Gala vērtējumu vērtības pa kursiem ārzemniekiem – Final grade values of foreign students by course*), the more the structure of these reports is alike (i.e. the major part of OLAP schema elements employed are the same). Naturally, lower value of similarity coefficient (as in #8: *Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem – Total monthly students’ final grade count of foreign students by course*, #9: *Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem – Total monthly students’ interim grade count of foreign students by course*, and #10: *Kopējais uzdevumu skaits mēnesī pa kursiem – Total monthly students’ task count by course*) means the opposite. Similarity values that are a little over the average (0.696 and 0.667) are represented by reports #4: *Gala vērtējumu skaits mēnesī pa kursiem – Total monthly students’ final grade count by course*, #5: *Kopējais vērtējumu skaits mēnesī pa kursiem – Total monthly students’ grade count by course*, #6: *Starpvērtējumu skaits mēnesī pa kursiem – Total monthly students’ interim grade count by course*, and #7: *Vērtējumu tipu sadalījums mēnesī pa kursiem – Monthly distribution of students’ grade types by course*.

### Examples of Recommendations in Semantic Mode

In *semantic mode* the semantic hot-start method for generation of recommendations is employed. In terms of this method, user preferences are stated explicitly by means of terms selected by user from glossaries and degrees of interest assigned to those terms. In Figure 6.2.5.3 several examples of terms are given: here a user selected *Ārzemnieks – Foreign student* from the glossary *Studiju process – Study process*, *Vērtējumu skaits – Number of grades* from the glossary *Vērtēšanas process – Assessment process*, and *Mēnesis – Month* from the glossary *Laiks – Time*.

When all the terms of interest are chosen, a user assigns the degree of interest (DOI) from the drop-down list (see Figure 6.2.5.4). Values of the DOI are expressed in numbers that may vary from 0 (not interested) to 100 (highly interested) with an interval of 5. Terms with DOI = 0 are not being saved to the user profile, however, this option is handy, if the user is no longer interested in one or another term and wishes to delete it from the profile.

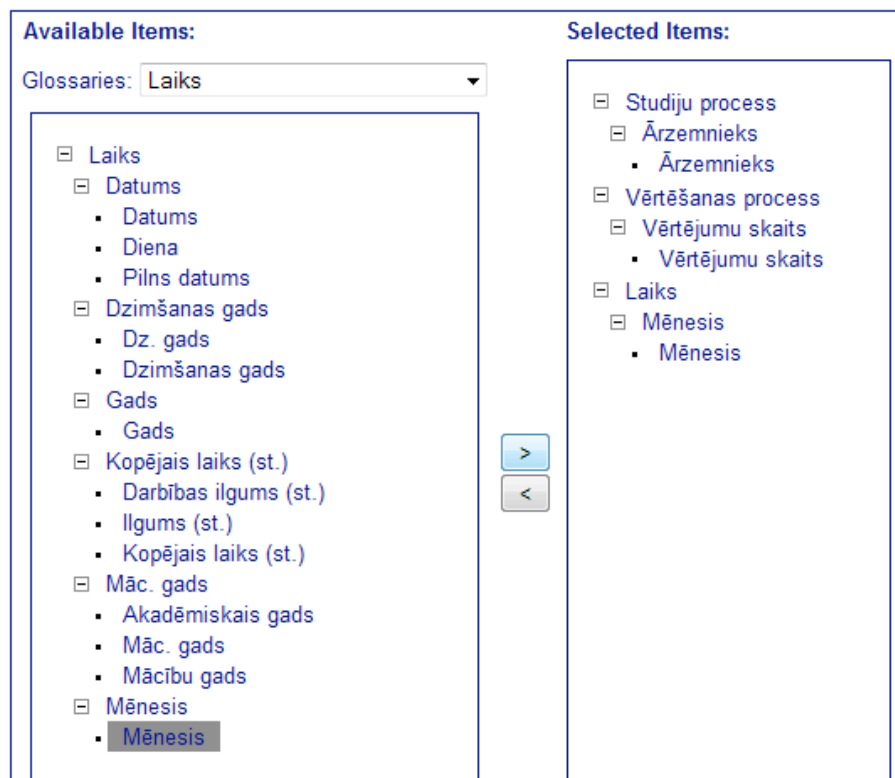


Fig. 6.2.5.3. Examples of terms selected by user in semantic mode

In Figure 6.2.5.4 a user states that he/she is highly interested in report about *Foreign students* with DOI = 95 (*Ārzemnieks*), moderately interested in reports containing *Number of grades* with DOI = 75 (*Vērtējumu skaits*), and less interested in reports containing data split by *Month* with DOI = 55 (*Mēnesis*). Recommendations in semantic mode are generated as soon as the user finishes editing his/her profile.

Glossary	Concept	Term	Degree of Interest
Laiks	Mēnesis	Mēnesis	55 ▾
Studiju process	Ārzemnieks	Ārzemnieks	95 ▾
Vērtēšanas process	Vērtējumu skaits	Vērtējumu skaits	75 ▾
<input style="margin-right: 10px;" type="button" value=" &lt; Back "/> <input style="margin-right: 10px;" type="button" value=" Finish "/> <input style="margin-right: 10px;" type="button" value=" Cancel "/>			

**Fig. 6.2.5.4.** Examples of the degrees of interest (DOIs) assigned to terms selected by user in semantic mode

The usage scenario in Figure 6.2.5.5 includes 10 recommendations generated for one of the users in semantic mode. As in user activity mode, recommendations are sorted by the similarity coefficient value in descending order: first, by the dimension-based similarity coefficient value, then, by the fact-based similarity coefficient, and finally, by aggregate function DOI.

**Vērtējumu grāmata - Gala un starpvērtējumu vērtības**

Gala vērtējumu vērtības pa kursiem	Starpvērtējumu vērtības pa kursiem	Gala vērtējumu vērtības pa kursiem ārzemniekiem	Starpvērtējumu vērtības pa kursiem ārzemniekiem
Page <input style="margin-right: 5px;" type="button" value=" &lt; "/> 1 / 5 <input style="margin-left: 5px;" type="button" value=" &gt; "/> <input style="width: 30px; border: 1px solid gray;" type="text" value="1"/> <input style="margin-left: 10px;" type="button" value=" Show "/> <span style="float: right; margin-right: 20px;"><input style="border: 1px solid gray;" type="button" value=" Hide Report Recommendations "/></span>			
Nr.	Recommendation by Semantic Meaning	Similarity	
1	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.247; 0.183	
2	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.247; 0.183	
3	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Gala vērtējumu vērtības pa kursiem ārzemniekiem</a>	0.183; 0.183	
4	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Starpvērtējumu vērtības pa kursiem ārzemniekiem</a>	0.183; 0.183	
5	<a href="#">Studentu sadalījums pa dzimumiem. Studentu dzimumi fakultātēs pa studiju veidiem - ārzemnieki</a>	0.164; 0.000	
6	<a href="#">Studentu sadalījums pa dzimumiem. Studentu dzimumi pa studiju veidiem un tem. jomām - ārzemnieki</a>	0.164; 0.000	
7	<a href="#">Atjaunojušies un 1. kursa atbirums. Atjaunojušies - ārzemnieki</a>	0.158; 0.000	
8	<a href="#">Atskaitītie studenti. Studentu atbirums pa fakultātēm un izglītības līmeņiem - ārzemnieki</a>	0.158; 0.000	
9	<a href="#">Studentu skaits uz 1. datumu. Studējošo skaits pa izglītības līmeņiem un programmām - ārzemnieki</a>	0.158; 0.000	
10	<a href="#">Vērtējumu grāmata - Vērtējumu skaits. Kopējais vērtējumu skaits mēnesī pa kursiem</a>	0.138; 0.183	

**Fig. 6.2.5.5.** An example of recommendations in semantic mode

The top reports are #1: *Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem – Total monthly students’ final grade count of foreign students by course* and #2: *Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem – Total monthly students’ interim grade count of foreign students by course*. Then follow reports #3: *Gala vērtējumu vērtības pa kursiem ārzemniekiem – Final grade values of foreign students by course* and #4: *Starpvērtējumu vērtības pa kursiem ārzemniekiem – Interim grade values of foreign students by course*, which have lower value of the dimension-based similarity coefficient. Report #10: *Kopējais*



vērtējumu skaits mēnesī pa kursiem – Total monthly students' grade count by course has the lowest value of the dimension-based similarity coefficient, but the value of the fact-based similarity coefficient is the same.

Note that in reports #5-#9 (#5: *Studentu dzīmumi fakultātēs pa studiju veidiem - ārzemnieki – Foreign student division by gender, faculty, and study type*, #6: *Studentu dzīmumi pa studiju veidiem un tem. jomām - ārzemnieki – Foreign student division by gender, study type, and thematic field*, #7: *Atjaunojušies - ārzemnieki – Foreign returning students*, #8: *Studentu atbirums pa fakultātēm un izglītības līmeņiem - ārzemnieki – Foreign student dropout by faculty and level of education*, and #9: *Studējošo skaits pa izglītības līmeņiem un programmām - ārzemnieki – Number of foreign students by level of education and study program*) fact-based similarity coefficient is equal to 0. This means that these reports do not contain measures that have an assigned DOI value (in this case, *Number of grades*) and do contain other measures, however, there still are some attributes from the user profile.

### 6.3. Summary of the Section

In this section an emphasis was placed on the methods for generation of recommendations on reports in a metadata-based reporting tool. All of these methods are included into the recommendation component of the reporting tool developed and put to operation in the University of Latvia. A model to expose main user and system activities was presented. Three implemented methods for generation of recommendations in OLAP reporting tool were proposed and illustrated by examples: hot-start method that defines user preferences implicitly for active users, cold-start method that defines user preferences implicitly for passive users, and semantic hot-start method that is aimed to define user preferences explicitly.

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).

There are four different modes in the recommendation component that exploit the above-mentioned methods. Namely, hot-start method is implemented in user activity mode, cold-start method – in report structure mode, a combination of hot-start and cold-start method – in automatic mode – where a system itself switches between the two modes depending on user activity, and semantic hot-start method – in semantic mode. Different usage scenarios of the recommendation component applied to real data warehouse reports on learning process

were presented including Top10 recommendations each. Main results presented in this section are reflected in the papers [KS11, SK11, KS12, Koz13].

Presumably, the cold-start method would be the most suitable for novice users, meanwhile, the two remaining methods – for advanced ones. Still, experimentation is needed to draw conclusions on these methods and their applicability. The next step is an evaluation of each method for report recommendation involving real users of the reporting tool. A detailed description of the experimentation and its results are presented in section 7.

## 7. AN EMPIRICAL STUDY TO EVALUATE METHODS FOR GENERATION OF RECOMMENDATIONS

### 7.1. The Intent of the Section

In this section a detailed plan and results of the experimentation in the OLAP reporting tool developed and put to operation in the University of Latvia are presented. The experimental study was performed in laboratory settings and was targeted to explore which of the methods for generating recommendations in the reporting tool has a deeper impact on users (i.e. produces more accurate recommendations). The main principles of how exactly the population were sampled and what restrictions were applied are discussed in this section. Precision/recall metrics were employed to gather user activity data from the log-table that was necessary to measure performance and then a comparative analysis by means of certain statistical tools was performed. Detailed results of the survey filled in by each of the experimentation participants are presented as chart graphs.

### 7.2. The Goal of the Experimentation and Research Questions

#### 7.2.1. The Goal of the Experimentation

A quantitative research was conducted through setting up an experiment. The goal template of the Goal/Question/Metric (GQM) method introduced by [Bas92] was adopted to formulate the goal of the experiment:

*Analyze methods for generation of report recommendations implemented in OLAP reporting tool for the purpose of evaluation with respect to their performance from the point of view of the researcher in the context of laboratory settings.*

#### 7.2.2. Research Questions

The four modes that exploit methods of generating report recommendations in the OLAP reporting tool, namely, user activity mode that employs hot-start method, report structure mode that employs cold-start method, automatic mode that employs a combination of cold-start and hot-start methods, and semantic mode that employs semantic hot-start method are described in section 6.2. To be more precise, there actually are five modes – four abovementioned modes that supply a user with recommendations and the one with no recommendations.

At this point the author faced a dilemma – whether to include a mode with no recommendations in the experimental study or not. Considering this question, Kitchenham et al. write “For laboratory studies, we can compare two defined technologies, one against the

other; but, it is usually not valid to compare using a technology with not using it" [KPP02]. The author partially agreed with [KPP02] and excluded the mode with no recommendations from the evaluation leaving only one question in user survey that tackles the mode with no recommendation (see question 14 in Appendix 5). However, it was important to let the user work with the reporting tool and complete a test task in the mode with no recommendations for multiple reasons: (i) to learn how to navigate and execute reports, (ii) to accumulate user activity during the session, and (iii) to be able to state his/her opinion about using recommendation modes.

Besides, an automatic mode is indeed produced by synthesizing two other modes, because in automatic mode the reporting tool switches between the report structure mode and user activity mode (see section 6.2 for details). So, it's not really a report recommendation mode of full value, and, in author's opinion, it should be eliminated from the experimental study.

There are two research questions (RQ1 and RQ2) to be covered in this empirical study, which are classified as descriptive-comparative questions [WHH03] and are the following:

RQ1 – *Which of the implemented modes (and its underlying methods) of generating report recommendations in the OLAP reporting tool – i.e. user activity, reports structure, or semantic mode – has a deeper impact on users?*

RQ2 - *Which of type of methods for gathering user preferences – implicit (implemented in user activity mode and reports structure mode) or explicit (implemented in semantic mode) – has a deeper impact on users?*

In terms of this section a mode has a *deeper impact* on a user (or, in other words, outperforms the other mode), if it produces recommendations with more accuracy (which can be measured, see sections 7.3.3. and 7.4.1.) and leads to completing the task using the recommendation component of the reporting tool extensively.

To be more specific, here a *task* is one of the exploratory tasks of equal complexity, which is assigned to a user in a certain recommendation mode. There are 4 tasks in each user group – students, academic staff, and administrative staff (see section 7.3.2. for details) – and each task consists of 4 subtasks. Each subtask implies some data to be found in terms of a single report. All subtasks are neither trivial, nor sophisticated, because in each of them a user has to be able to understand and find the necessary reports and data, change report settings (e.g. parameters and page items), and switch between report pages. Therefore, report recommendations add some value to the process of user interaction with the reporting tool.

First, users complete a test task in the mode with no recommendations, then – the 1st task in report structure mode, then – the 2nd task in the semantic mode, and finally, they complete the 3rd task in user activity mode (in this mode all user activity during the experimental session is analyzed). The order of these tasks is the same for all users, however, tasks vary depending on the user group and rights on reports. All user tasks for student, academic staff, and administrative staff user groups are available in Appendices 1, 2, and 3 respectively.

### **7.2.3. *Philosophical Stance***

*Positivism* is the philosophical stance to be adopted as it is most closely associated with experiments according to [WHH03]. The answers to the research questions would come from an experiment executed under laboratory conditions.

## **7.3. Research Methodology**

While completing the empirical study, the author consulted the guidelines for conducting an experimental study [Bas92, KPP02, WHH03, ESSD08]. The design of the procedure for running the experiment is fixed, primary data is quantitative, and primary objective is explanatory.

### **7.3.1. *Context of the Experimental Study***

Here the context of the experiment is interpreted as background information about the industrial circumstances described by [KPP02], which needs to be defined to emphasize that the results of the experimentation are valid in certain industrial circumstances and cannot be generalized irrespective of the context. Thus, the following factors [KPP02] are identified:

- The industry in which products are used: Educational Institution (the University of Latvia).
- The nature of the software development organization: The tool was developed at the Faculty of Computing of the University of Latvia in terms of the ESF project, which means that the software development organization may be classified as an in-house software supplier.
- The skills and experience of software staff with the reporting tool: The subjects (or participants of the experiment) classify themselves according to their experience with any kind of reporting tools, which is defined by the number of times they used it: novice (i.e. has never/several times used any reporting tool), advanced user (i.e. has

occasionally used some reporting tool), and expert (i.e. has regularly used some reporting tool and/or has skills in reporting tool/report development).

- The type of software products used: A tool for administrating, designing, modifying, and executing data warehouse reports.
- The software processes being used: Software operation process (as the subjects were required to employ OLAP reporting tool in different modes of generating report recommendations).

### 7.3.2. *Subjects*

An experiment was conducted with a certain set of report data on user interaction with Moodle course management system (referred as Moodle or Moodle CMS) and study process in the University of Latvia. By the time when the experiment took place, 70 reports had been available for the subjects.

The population for the experiment consists of dedicated and motivated participants (or subjects) who are related to the University of Latvia and who are interested in the reports. Moreover, either the subjects are Moodle users and are directly involved in the study process (for instance, students and academic staff) or they are interested in an overview of user activity in Moodle and study process (for instance, administrative staff).

However, one of the shortcomings is that even though each course of any study program in the University of Latvia has a corresponding e-course in Moodle CMS, Moodle is not actively employed in all faculties of the University of Latvia and a part of the e-courses barely has any content. Thus, the scope of participants narrows to those who are active users of Moodle CMS, namely, representatives of the Faculty of Computing, IT and Academic department. The author herself did not take part in the experimentation as a subject.

In statistics a rule of thumb (suggested by Roscoe [Ros75]) is that in experimental research samples of 30 or more are recommended, which is why there are 30 participants of the experimental study. It was decided to split the subjects in 3 groups (or blocks) according to the distinction in rights on report data, thus, making the population more diverse and closer to the real-life circumstances. The 3 groups are the following:

- *Students*. The main consumers of the Moodle e-course content. In the reporting tool they would be interested to get detailed data that mostly describes them, for example, their personal grades and activities in Moodle and study process, average grades in the courses that they take or are planning to take in the future, quantity of tasks in these courses to evaluate the approximate workload, etc. This group consists of 10 subjects

and includes bachelor, master, and PhD students. Subjects of this group had rights to execute 65 reports.

- *Academic staff.* The ones who monitor the study process and participate in creating content for Moodle CMS (e.g. lecturers, professors). In the reporting tool they would be interested to get general data, for example, on student progress in their courses, on how their e-courses in Moodle CMS differ from others, etc. This group consists of 8 subjects and includes lecturers, associate professors, and professors. Subjects of this group had rights to execute 65 reports.
- *Administrative staff.* The ones who monitor study process and make decisions on how to invest in the study process (e.g. department directors). In the reporting tool they would be interested to get data generalized on the level of faculty or study program, for example, on usage of Moodle gradebook tool by professors and students, on total number of students who joined the University and graduated, etc. This group consists of 12 subjects and includes department directors and deputy directors, program directors, study methodologist, dean of the faculty of Computing, methodologist at the Academic department, and PR specialist of the faculty of Computing. Subjects of this group had rights to execute 70 reports.

### 7.3.3. Variables

Table 16 lists three values of one independent variable, which is recommendation mode. Each mode ( $M_{UA}$ ,  $M_{RS}$ , and  $M_S$ ) in fact has an underlying method of generating report recommendations in the OLAP reporting tool (hot-start, cold-start, and semantic hot-start respectively).

**Table 16. Independent variables**

<i>Variable Name</i>	<i>Variable Value</i>	<i>Abbr.</i>	<i>Class</i>	<i>Entity</i>
Mode of generating report recommendations	User activity mode	$M_{UA}$	Method	Hot-start method
	Report structure mode	$M_{RS}$	Method	Cold-start method
	Semantic mode	$M_S$	Method	Semantic hot-start method

The Precision/Recall metrics is widely used in information retrieval as well as in the field of recommender systems [STL11, SG11, SG13]. Often a 2x2 confusion matrix (or contingency table) is built to present a binary classification of some prediction problem and to analyze predicted and actual outcomes. In general, there are four possible outcomes from a binary classifier:

- *True positive* – if both the prediction outcome and the actual value are  $p$  (*correct values are classified correctly*);
- *False positive* – if the prediction outcome is  $p$ , but the actual value is  $n$  (*incorrect values are erroneously classified as correct*);
- *False negative* – if the prediction outcome is  $n$ , whereas the actual value is  $p$  (*correct values are erroneously classified as incorrect*);
- *True negative* – if both the prediction outcome and the actual value are  $n$  (*incorrect values are classified correctly*).

In the context of the recommendation component of the reporting tool the characteristics of all four outcomes need to be explained at greater length. Suppose that throughout the whole session of user's interaction with the reporting tool one can detect a set of reports that have been relevant for the user in terms of providing data of interest (RL) and a set of ones that haven't been (NRL). Meanwhile, a user has two options while exploring reports in order to collect necessary data – whether to use a recommendation component or not. In this particular case, the characteristics of the possible outcomes are defined as follows:

- *True positive (TP)* – the number of relevant reports that the user examined by means of hitting the link in the recommendation component (*reports belonging to RL set were correctly labeled as relevant*);
- *False positive (FP)* – the number of irrelevant reports in the recommendation component (*reports belonging to NRL set were mistakenly labeled as relevant*);
- *False negative (FN)* – the number of relevant reports that the user examined not following\* the recommendation link (*reports belonging to RL set were mistakenly labeled as irrelevant*);
- *True negative (TN)* – the number of irrelevant reports that were not displayed as recommendations during the session (*reports belonging to NRL set were correctly labeled as irrelevant*).

An appropriate metrics to evaluate the performance of the reporting tool is *classification accuracy metrics*, because according to an overview of metrics for recommender system evaluation presented in [STL11], it “measures the amount of correct and incorrect classifications as relevant or irrelevant items that are made by the recommender

---

\* For instance, if the user ignored the recommendations, but then, while looking through the report list, came across the report from a recommendation list and decided to explore it.



system and are therefore useful for user tasks such as finding good items”. The exact rating or ranking of items is ignored, as the measure of interest is either correct or incorrect classification. Although report recommendations in the recommendation component of the reporting tool are sorted by similarity coefficient value, evaluation of the order of recommendations is of minor importance. It is so, because in terms of exploratory tasks a user may sometimes find useful report recommendations that not necessarily are in the beginning of the recommendation list. Moreover, during the experimentation process, several users admitted that they start reading a recommendation list not from the top, but from the bottom.

**Table 17. Dependent variables**

<i>Variable Name</i>	<i>Abbr.</i>	<i>Entity</i>	<i>Scale Type</i>	<i>Range</i>	<i>Counting Rule</i>
True Positive	TP	The number of relevant reports that the user executed by means of hitting a recommendation link	Ratios	$N \cup \{0\}$	Sum
False Positive	FP	The number of irrelevant reports in the recommendation component	Ratios	$N \cup \{0\}$	Sum
False Negative	FN	The number of relevant reports that the user executed not following a recommendation link	Ratios	$N \cup \{0\}$	Sum
Precision	P	The ratio of reports accessed by user via recommendation link and executed to the total number of relevant and irrelevant reports in the recommendation component	Ratios	$Q, [0; 1]$	$TP / (TP + FP)$
Recall	R	The ratio of reports accessed by user via recommendation link and executed to the total number of reports classified as relevant and executed	Ratios	$Q, [0; 1]$	$TP / (TP + FN)$
F <sub>1</sub> -measure	F <sub>1</sub>	The standartized harmonic mean of precision and recall	Ratios	$Q, [0; 1]$	$2 * P * R / (P + R)$

Table 17 demonstrates dependent variables required to measure the performance of the methods for generation of recommendations. All dependent variables are perceived as resources, because user activity logged during the experiment indeed is a resource of data for future analysis. The values of TP, FP, and FN are expressed as whole numbers starting with 0. The values of TN do not characterize the usage of recommended reports; moreover, TN does not affect Precision (P) and Recall (R) and is not needed in further evaluation, therefore, it is

excluded from the Table 17. The value of  $P$  is the ratio of reports accessed by a user via recommendation link and executed to the total number of relevant and irrelevant reports in the recommendation component. The value of  $R$  is the ratio of reports to execute that were accessed by user via recommendation link and executed to the total number of reports classified as relevant and executed by user (i.e. recommendations that were accessed either by following or not following a recommendation link). The values of  $P$  and  $R$  are rational numbers on the segment  $[0; 1]$ . One more variable is  $F_1$ -measure (or  $F_1$ -score) is a measure of test's accuracy that combines precision and recall into a single value by calculating different types of means of both metrics [STL11]. The  $F_1$ -measure is calculated as the standartized harmonic mean of precision and recall, where the best  $F_1$ -measure has its value at 1 and worst – at 0.

The quantity of recommendations being shown to the user together with the report (i.e. *TopN*) may vary. Being guided by the authors [SKKR00] who calculated the optimal length of the recommendation list, which is 10, in terms of this experiment  $N$  is set to 10.

To learn the values of all dependent variables acquired by the end of the experimentation and its application when analyzing the performance of recommendation generation modes, see section 7.4.1.

#### 7.3.4. *Design Principles*

It is common for guidelines on conducting an empirical research [e.g. KPP02, WHH03] to point out that samples from the population should be selected randomly to provide the most convincing results of the experimentation. In the same time, Wohlin et al. state “Ideally, it would be possible to randomly choose a sample from the population to include in the study, but this is for obvious reasons mostly impossible. Often, we end up trying to determine to which population we can generalize the results from a certain set of participants” [WHH03].

The population was chosen randomly, but with several restrictions (exclusion criteria): (i) a subject should have been a dedicated Moodle user or directly involved in the study process, (ii) a subject should have been interested in taking part in the experimentation (bearing in mind that the whole experimentation process might take more than 1 hour per subject), and (iii) if the subject was a representative of more than one group, then he/she could take part in the experiment only once. That way, the design principle of randomization was applied with restrictions.

The number of the representatives of each group is uneven. Administrative staff is the group of users most interested in reports and its data, because often they are the ones to make

decisions – for that reason in terms of the experimentation administrative staff group has the largest number of participants. Also, the overall number of students is higher than that of the academic staff. Thus, the number of subjects in each group is unequal too: students – 10, academic staff – 8, and administrative staff – 12 participants. The design principle applied was blocking on rights (students/academic staff/administrative staff, see section 7.3.2) or blocking on experience with reporting tools (novice/advanced users & experts, see Appendix 6).

The subjects had to perform 4 different tasks consecutively: one task not applying any recommendation mode, and 3 tasks applying a certain recommendation mode – one task in user activity mode ( $M_{UA}$ ), one task in report structure mode ( $M_{RS}$ ), and one task in semantic mode ( $M_S$ ). In literature such approach where each subject uses all treatments is classified as “within subjects design” [ESSD08]. These tasks differ in each of 3 groups of subjects. A subject had to complete each task in approximately 20 minutes time (estimated time – about 1 hour 20 minutes in total) without any interaction with other participants of the experiment (individually). The time required for completing each task depended on individual abilities of each subject in particular (for example, experience in reporting tools, knowledge of data domain, etc.), which is why there was no strict time frame. Each task was considered to be completed, when a subject had completed all 4 subtasks. Average time for a participant to complete all 4 tasks was 1 hour 30 minutes.

One may raise a concern over the fact that a subject might have learned how to use the reporting tool and the data gathered in the reports. It seemed unlikely to fully prevent a subject from learning, however, to mitigate learning bias, 3 out of 4 tasks (test task, 1st, and 2nd) covered reports from different workbooks, thus, making a subject explore new reports and data. In the 3rd task the reports to be found were either previously explored by the subject or similar to the explored ones, because this was the only way to test recommendations in user activity mode based on user activity during the session. Moreover, blind allocation of materials [KPP02] took place meaning that the subjects were distributed to one of three groups without actually knowing that such division exists.

After completing 1st–3rd task each user had to fill in a survey with multiple choice questions on each of the tasks (see full user survey in Appendix 5). The questions touched upon task clarity and complexity as well as if the recommendations were helpful and if the user had mostly used Top3 recommendations. In general questions users: (i) themselves stated their experience with reporting tools, (ii) compared task completion in any of the recommendation mode (1st–3rd task) with that without any recommendation mode (test task), (iii) stated the task(s) in which they used recommendation component most of all, and (iv)

stated the task(s) where they have received the most precise recommendations. Also, users could leave their comments in free form in the end of the survey. This way, user feedback to supplement experimentation results, subjective impressions on recommendation mode usage, and suggestions on what to improve in the reporting tool were collected.

### **7.3.5. Conducting the Experiment and Data Collection**

Subjects consented to participate in the study being informed that none of the identities would be disclosed when reporting the results. Then, during the individual meeting he/she was given an oral explanation considering the whole process of the experimentation as well as the data about the subject that was going to be collected and used to perform analysis and prepare summary of the study. The author demonstrated to the subject how to use the tool. Also, each participant was given a manual (closely related to the demo) with the necessary information on how to use the tool, i.e. execute reports and switch between recommendation modes (see Appendix 4).

Particular logging procedures had been added to the source code of the reporting tool to capture each click of the subject and characteristics associated with it (e.g. report ID, user ID, mode ID, current page loaded, button pressed, parameters entered, recommendation chosen, etc.) by inserting a new record into the log-table. To keep track of the recommendation component usage, there is a flag that indicates with 1 or 0, whether a subject has executed the report by hitting a recommendation link or not. This way, it was possible to analyze user activity and to check what reports a particular user had executed, which of them were part of the recommendation set and which were not, whether the user employed the recommendation component or not. The data on TP, FP, FN, and derived measured as P, R, and  $F_1$ -measure was gathered and summed up.

The general check to ensure the process and quality of data collection is essential. It included such steps as:

- To verify that the timestamps of the actual beginning and the end of the experiment session correspond to those in log-table;
- To check whether there were errors fixed while collecting data into the log-table during each session, and if so, then learn and treat the adverse factors that caused errors;
- To check if there actually is data in the log-table for each session, and it is not empty for whatever reason;

- To verify if the subjects have completed all 4 tasks and have filled in the survey and provided answers to all of the survey questions.

In [KPP02] the authors advise to record data about subjects who drop out from the studies. The author documented the data on subjects (for instance, user group, the reason of quitting the study) who dropped out in the very beginning of the study or abandon the experiment before completing all tasks.

In total there were 3 subjects (2 from student and 1 from administrative staff user group) who dropped out in the very beginning of the study: 2 subjects could not participate because of being abroad and 1 subject could not participate because of being ill. All the subjects who dropped out were substituted with other participants with the same level of rights and experience, so that the total number of subjects did not fall short of 30.

## 7.4. Experimentation Results

### 7.4.1. Results of the Log-table Analysis

After the data from all the subjects had been collected and approved for further research, an offline analysis began.

The tasks were assigned to 30 participants, since it is the minimum number of participants suggested by [Ros75] for an experimental study. Taking this fact into consideration, the author puts aside a thought about analyzing the interaction between the reporting tool and each block of participants separately (split by their rights in the reporting tool, see section 7.3.4) when calculating P, R, and F<sub>1</sub>-measure values acquired from the users while working in each of three recommendation modes (M<sub>UA</sub>, M<sub>RS</sub> and M<sub>S</sub>).

All values of dependent variables – TP, FP, FN, Precision (P), Recall (R), and F<sub>1</sub>-measure – gained from experimental tasks completed in report structure (M<sub>RS</sub>), semantic (M<sub>S</sub>), and user activity (M<sub>UA</sub>) modes are reported in Tables 18 – 20 respectively.

In [KPP02] it is advised not to ignore outliers. To identify outliers, one of the popular ways to test the extreme values – Grubb's test – was applied. An explanation of the principles of Grubb's test is found at GraphPad<sup>4</sup> statistics guide. It is also possible to determine whether one of the values is a significant outlier from the rest by the use of GraphPad QuickCalcs<sup>5</sup>. Sometimes a significant outlier may indicate an error (for example, age value is higher than 150, etc.), however, it may also mean an interesting or exceptional case that requires an explanation, and such values should not be removed from the data set.

<sup>4</sup> GraphPad statistics guide available at: <http://www.graphpad.com/guides/prism/6/statistics/>

<sup>5</sup> GraphPad QuickCalcs available at: <http://graphpad.com/quickcalcs/Grubbs1.cfm>

**Table 18. Dependent variable values in report structure mode**

<i>Mode</i>	<i>Sample Nr.</i>	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>P</i>	<i>R</i>	<i>F<sub>1</sub>-measure</i>
<i>M<sub>RS</sub></i>	1	5	5	0	0.500	1.000	0.667
	2	3	7	0	0.300	1.000	0.462
	3	3	7	0	0.300	1.000	0.462
	4	3	7	0	0.300	1.000	0.462
	5	4	5	1	0.444	0.800	0.571
	6	2	7	1	0.222	0.667	0.333
	7	3	7	0	0.300	1.000	0.462
	8	3	7	0	0.300	1.000	0.462
	9	3	7	0	0.300	1.000	0.462
	10	2	6	2	0.250	0.500	0.333
	11	6	3	1	0.667	0.857	0.750
	12	2	7	1	0.222	0.667	0.333
	13	3	7	0	0.300	1.000	0.462
	14	2	8	0	0.200	1.000	0.333
	15	3	7	0	0.300	1.000	0.462
	16	4	6	0	0.400	1.000	0.571
	17	6	4	0	0.600	1.000	0.750
	18	5	5	0	0.500	1.000	0.667
	19	3	7	0	0.300	1.000	0.462
	20	4	5	1	0.444	0.800	0.571
	21	3	7	0	0.300	1.000	0.462
	22	3	7	0	0.300	1.000	0.462
	23	3	7	0	0.300	1.000	0.462
	24	7	3	0	0.700	1.000	0.824
	25	3	6	1	0.333	0.750	0.462
	26	3	7	0	0.300	1.000	0.462
	27	3	7	0	0.300	1.000	0.462
	28	3	7	0	0.300	1.000	0.462
	29	3	7	0	0.300	1.000	0.462
	30	3	7	0	0.300	1.000	0.462
<b>Mean values</b>					<b>0.353</b>	<b>0.935</b>	<b>0.500</b>

**Table 19. Dependent variable values in semantic mode**

<i>Mode</i>	<i>Sample Nr.</i>	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>P</i>	<i>R</i>	<i>F<sub>1</sub>-measure</i>
<i>M<sub>s</sub></i>	1	5	5	0	0.500	1.000	0.667
	2	4	6	0	0.400	1.000	0.571
	3	3	7	0	0.300	1.000	0.462
	4	5	5	0	0.500	1.000	0.667
	5	4	6	0	0.400	1.000	0.571
	6	3	7	0	0.300	1.000	0.462
	7	0	7	3	0.000	0.000	0.000*
	8	3	7	0	0.300	1.000	0.462
	9	3	7	0	0.300	1.000	0.462
	10	3	7	0	0.300	1.000	0.462
	11	4	5	1	0.444	0.800	0.571
	12	3	7	0	0.300	1.000	0.462
	13	2	8	0	0.200	1.000	0.333
	14	3	7	0	0.300	1.000	0.462
	15	3	7	0	0.300	1.000	0.462
	16	3	5	2	0.375	0.600	0.462
	17	5	5	0	0.500	1.000	0.667
	18	3	7	0	0.300	1.000	0.462
	19	3	7	0	0.300	1.000	0.462
	20	3	7	0	0.300	1.000	0.462
	21	2	8	0	0.200	1.000	0.333
	22	2	8	0	0.200	1.000	0.333
	23	3	7	0	0.300	1.000	0.462
	24	6	4	0	0.600	1.000	0.750
	25	4	6	0	0.400	1.000	0.571
	26	5	5	0	0.500	1.000	0.667
	27	2	8	0	0.200	1.000	0.333
	28	5	5	0	0.500	1.000	0.667
	29	2	8	0	0.200	1.000	0.333
	30	2	8	0	0.200	1.000	0.333
<b>Mean values</b>					<b>0.331</b>	<b>0.947</b>	<b>0.479</b>

**Table 20. Dependent variable values in user activity mode**

<i>Mode</i>	<i>Sample Nr.</i>	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>P</i>	<i>R</i>	<i>F<sub>1</sub>-measure</i>
<i>M<sub>UA</sub></i>	1	4	6	0	0.400	1.000	0.571
	2	0	10	0	0.000	0.000	0.000
	3	3	6	1	0.333	0.750	0.462
	4	4	6	0	0.400	1.000	0.571
	5	1	9	0	0.100	1.000	0.182
	6	4	6	0	0.400	1.000	0.571
	7	2	8	0	0.200	1.000	0.333
	8	2	8	0	0.200	1.000	0.333
	9	3	7	0	0.300	1.000	0.462
	10	3	7	0	0.300	1.000	0.462
	11	2	5	3	0.286	0.400	0.333
	12	1	9	0	0.100	1.000	0.182
	13	3	7	0	0.300	1.000	0.462
	14	2	7	1	0.222	0.667	0.333
	15	3	7	0	0.300	1.000	0.462
	16	2	8	0	0.200	1.000	0.333
	17	0	10	0	0.000	0.000	0.000
	18	3	7	0	0.300	1.000	0.462
	19	3	7	0	0.300	1.000	0.462
	20	1	9	0	0.100	1.000	0.182
	21	1	9	0	0.100	1.000	0.182
	22	2	8	0	0.200	1.000	0.333
	23	2	8	0	0.200	1.000	0.333
	24	4	6	0	0.400	1.000	0.571
	25	3	7	0	0.300	1.000	0.462
	26	1	9	0	0.100	1.000	0.182
	27	2	8	0	0.200	1.000	0.333
	28	2	8	0	0.200	1.000	0.333
	29	2	8	0	0.200	1.000	0.333
	30	2	8	0	0.200	1.000	0.333
<b>Mean values</b>					<b>0.228</b>	<b>0.894</b>	<b>0.352</b>

Outlier tests with GraphPad QuickCalcs for F<sub>1</sub>-measures acquired in each of the recommendation modes showed that there are no significant outliers in M<sub>RS</sub> and M<sub>UA</sub>, whereas there is 1 significant outlier in M<sub>S</sub>, is marked with ‘\*’ in Table 19 (sample nr. 7). Here a subject ignored the recommendations and found the relevant reports (which were also in the recommendation list) by browsing the reporting tool.

Now, let’s formulate the null hypotheses derived from the RQ1 and RQ2:



- $H_{01}$ : There is no significant difference in the performance of generating recommendations in mode  $M$  and in the remaining modes, where  $M \in \{M_{RS}, M_S, M_{UA}\}$ ;
- $H_{02}$ : There is no significant difference in the performance of generating recommendations between modes employing methods that gather user preferences implicitly and the one that gathers it explicitly.

Before applying any test on  $F_1$ -measure values, one has to clarify, if each of the  $F_1$ -measure values acquired in any of the recommendation is normally distributed or not. The Shapiro-Wilk normality test can be easily performed online<sup>6</sup> by pasting a set of values. Like in many statistical tests, the P-value, which in the field of statistics is referred as statistical significance, is calculated and compared. Standard deviation is calculated too.

As the results of Shapiro-Wilk normality test show, the  $F_1$ -measure data in each of the recommendation modes is not normally distributed. This means that to test the above-mentioned null hypotheses one should use, for example, an online Mann-Whitney test<sup>7</sup>, which is suitable for non-normally distributed data.

**Table 21. Results of the Mann-Whitney test**

<i>Mode for <math>F_1</math>-measure values in Sample A</i>	<i>Mode for <math>F_1</math>-measure values in Sample B</i>	<i>U</i>	<i>P-value (approx.)</i>	<i>Result</i>	<i>Accept / Reject Null Hypothesis?</i>
$M_{RS}$	$M_S$	467.0	0.806782	The two samples are not significantly different ( $P \geq 0.05$ )	$H_{01}$ accepted
$M_{RS}$	$M_{UA}$	680.0	0.000566	The difference between the two samples is highly significant ( $P < 0.001$ )	$H_{01}$ rejected
$M_S$	$M_{UA}$	654.0	0.002316	The two samples are significantly different ( $P < 0.01$ )	$H_{01}$ rejected
$M_S$	$M_{RS}$ and $M_{UA}$	600.0	0.026018	The difference between the two samples is marginally significant ( $P < 0.05$ )	$H_{02}$ rejected

Then, to either accept or reject  $H_{01}$ , 3 pairwise comparisons of  $F_1$ -measure values have to be made:  $F_1$ -measure values in (i)  $M_{RS}$  and  $M_S$ , (ii)  $M_{RS}$  and  $M_{UA}$ , and (iii)  $M_S$  and  $M_{UA}$ .

<sup>6</sup> Shapiro-Wilk normality test available at: <http://sdittami.altervista.org/shapirotest/ShapiroTest.html>

<sup>7</sup> Mann-Whitney test available at: <http://elegans.som.vcu.edu/~leon/stats/utest.html>

When the calculated two-tailed P-value is less than 0.05 ( $P < 0.05$ ), the conclusion is that the two sets of  $F_1$ -measure values in question are significantly different.

The process of testing  $H_{02}$  is almost similar to the abovementioned. The only peculiarity is that one should get the mean of  $F_1$ -measure values referring to the modes that employ implicit user preferences (i.e.  $M_{UA}$  and  $M_{RS}$ ). The values of  $F_1$ -measure in a mode employing explicit user preferences are those in  $M_S$ . The resulting P-value would indicate, whether  $H_{02}$  should be supported or rejected.

Table 21 gives a summary of Mann-Whitney test and demonstrates calculated P-values and states the difference. The conclusions are as follows:

- There is no significant difference in performance of the recommendation component of the reporting tool in report structure ( $M_{RS}$ ) and semantic ( $M_S$ ) modes;
- Meanwhile, the recommendation component in report structure or in semantic mode outperforms that in user activity ( $M_{UA}$ ) mode;
- There a marginally significant difference in the performance of generating recommendations between modes that gather user preferences implicitly and the one that gathers it explicitly.

The results of the log-table analysis show that report structure and semantic modes (with a little difference in scores) produce the most relevant report recommendations for users regardless of their experience or belonging to a certain user group, whereas the lower number of relevant recommendations appears in user activity mode. Recommendations in user activity mode are affected by report execution, which does not always reflect user interest, especially, in a short period of time (as it was in terms of the experimentation). However, it would be valuable to see how the results of the recommendation mode usage acquired from the log-table correlate with user impressions provided directly.

Participants of the experimentation were asked to fill in the user survey after they had finished the practical part of the experimentation. The analysis of user experience and their preferred recommendation mode(s) reflected in the user survey and feedback will follow in the next section.

#### **7.4.2. Results of the User Survey Represented Graphically**

Alternatively, the author would like to summarize the information acquired about the subjects from user survey during the process of experimentation and represent it through the instrumentality of charts. The charts in the form of stacked columns reflect percentage of answers to each question of the survey in each block of participants split by their rights (i.e. student/academic staff/administrative staff user groups).

The survey (see Appendix 5) supplements the experimentation results collected from user activity history. The survey sampling method is cluster-based sampling as surveying individuals belong to three different groups: administrative staff, academic staff, and students. Those groups do not intersect as an individual can take part in the experimentation and survey as a representative of only one group.

All survey results are represented as stacked column graphs with indicated percentage values, where the total number of survey respondents (or experimentation participants) is 30 and is equal to 100%. All of the experimentation participants have provided answers to all of the questions in the survey.

The response results to survey questions 1, 5, and 9 are demonstrated in Figure 7.4.2.1 graphs (a), (b), and (c) respectively. The target of all three questions is to get an evaluation of the complexity of the experimentation task in each of three recommendation modes. As it is seen from the graph (a) in Figure 7.4.2.1, the 1st task is qualified as “Easy” mostly for representatives of the student user group (20%), whereas the majority of other user group members – academic and administrative staff – tend to rate it as “Average” – 16.67% and 20% respectively. There is a minor difference in “Easy” and “Average” overall rating, which makes up 3.33%. The 2nd task (see Figure 7.4.2.1 graph (b)) is rated as “Average” by all user groups. Besides, this is the only task with “Very hard” rating (6.66% in total). Mainly, the explanation for such a rating is that the 2nd task had to be completed in semantic mode, where a user needed to learn how to fill in the user profile – define preferences and assign DOIs. This required an extra effort from a user, thus, leaving an impression that the 2nd task seemed harder than the others. The 3rd task (see Figure 7.4.2.1 graph (c)) is perceived easier than others, because “Easy” rating value significantly exceeds other rating values in all 3 user groups. Moreover, it is the only task that has a “Very easy” rating (3.33%). The explanation for such a result is that by the end of the experimentation a user already knows well how to deal with the reporting tool and recommendations.

The response results to survey questions 2, 6, and 10 are demonstrated in Figure 7.4.2.2 graphs (a), (b), and (c) respectively. The target of all three questions is to get an evaluation of the clarity of the experimentation task in each of three recommendation modes. The 1st (see Figure 7.4.2.2 graph (a)) and the 3rd task (see Figure 7.4.2.2 graph (c)) have a very similar division of ratings in all three groups between “Clear” – 63.33% and 60% in total, “Mostly clear” – 33.33% and 36.67% in total, and “Mostly confusing” – 3.33% and 3.33% in total in the 1st and the 3rd task respectively. In the 2nd task (see Figure 7.4.2.2 graph (b)) “Mostly clear” prevails over other rating values and “Mostly confusing” rating value has a total of 10% in student and administrative staff group, because (i) in the 1st

subtask of the 2nd task (see Appendices 1-3) some participants asked to give an explanation to “Thematic field” term as they have never used that term before, and (ii) in the 2nd subtask of the 2nd task (see Appendices 1-3) 2 data values were required to be found (Faculty and the Number of students), which confused some users that wanted to acquire these values not from a single reports (as it was initially requested in the task description), but from 2 reports.

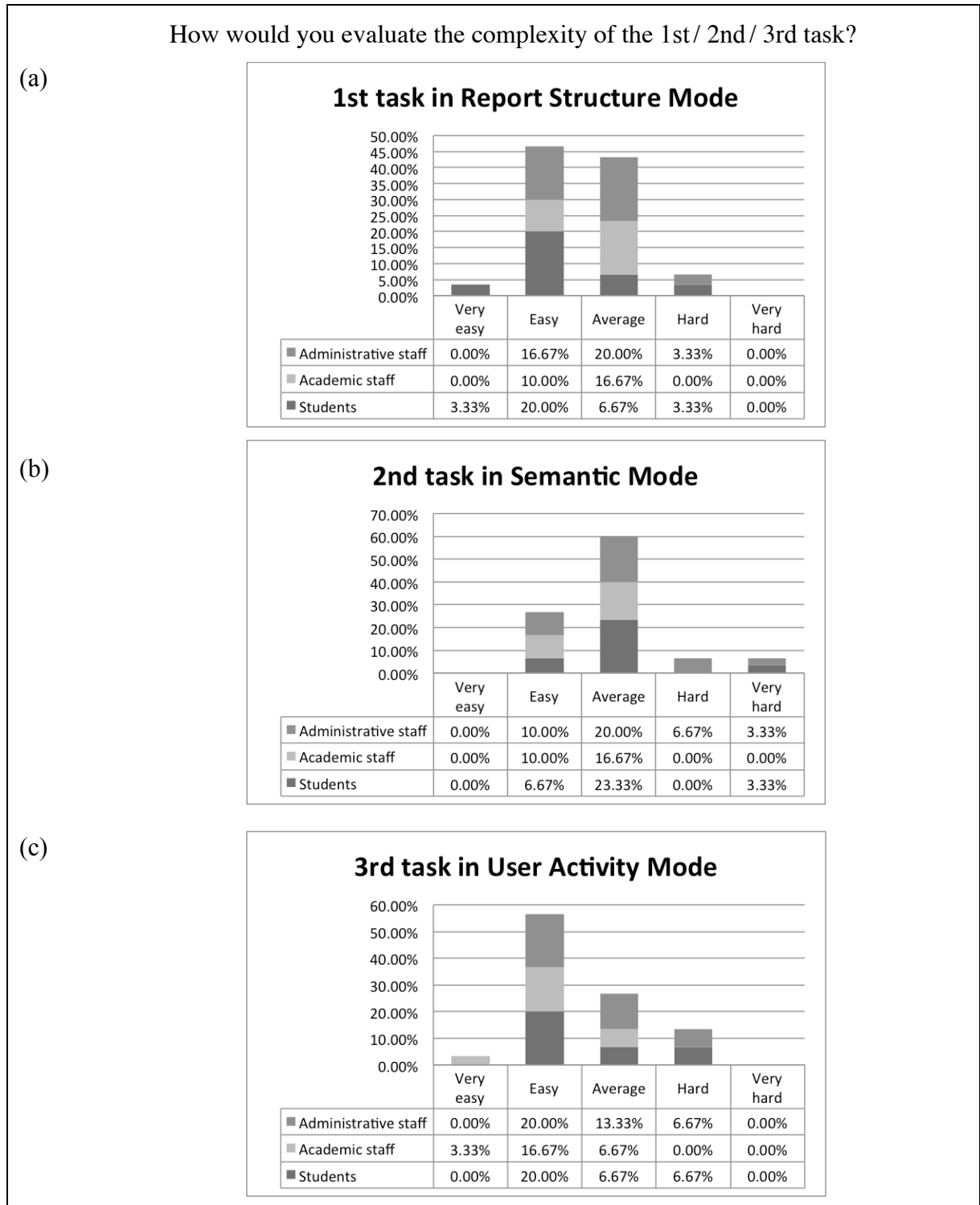


Fig. 7.4.2.1. Response results to survey questions 1, 5, and 9

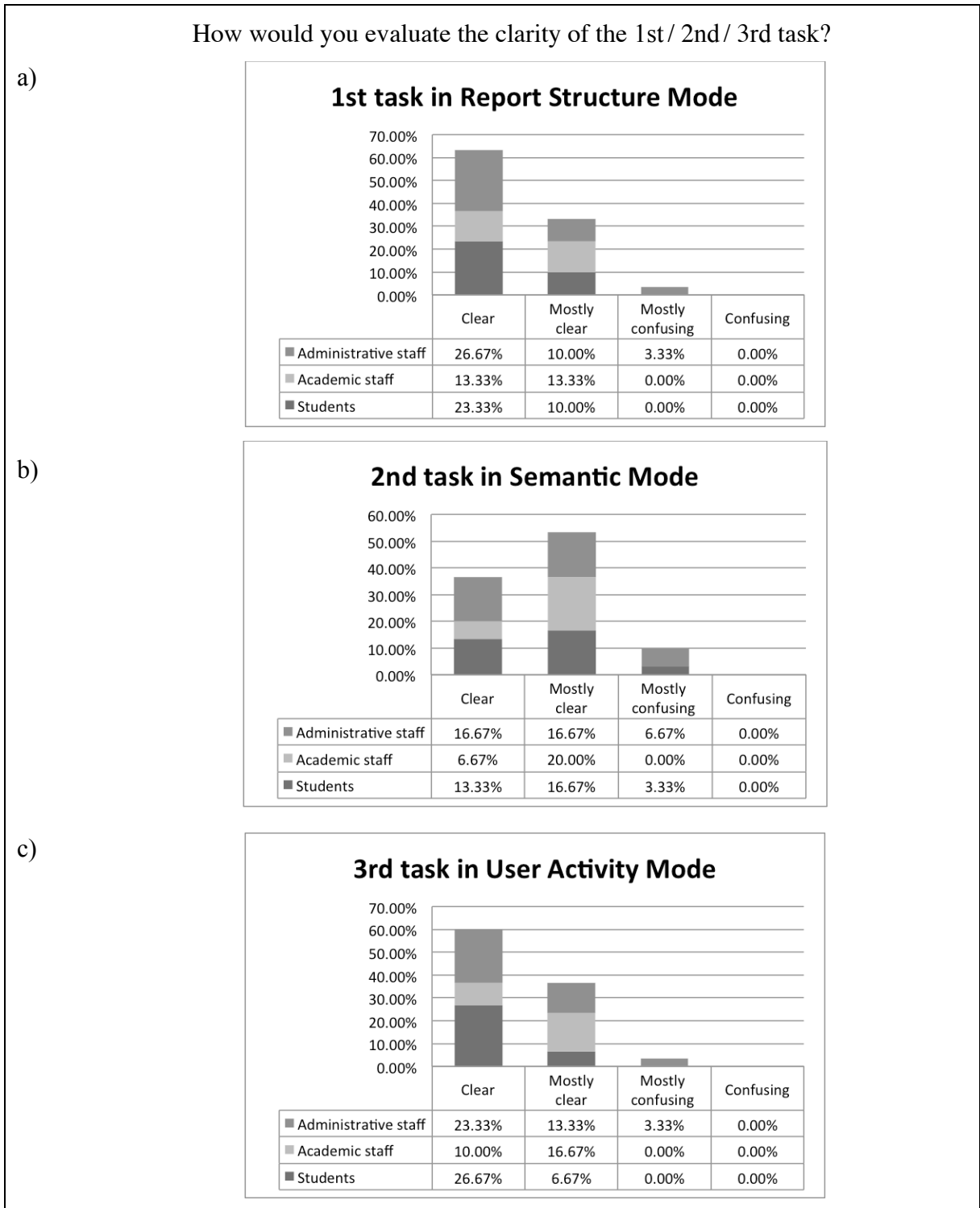


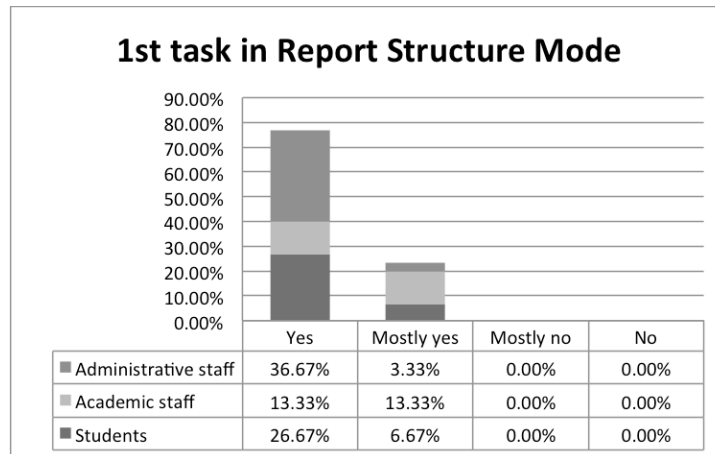
Fig. 7.4.2.2. Response results to survey questions 2, 6, and 10

The response results to survey questions 3,7, and 11 are demonstrated in Figure 7.4.2.3 graphs (a), (b), and (c) respectively. The target of all three questions is to clarify, whether the report recommendations helped each of the participants complete the experimentation task in each of three recommendation modes.

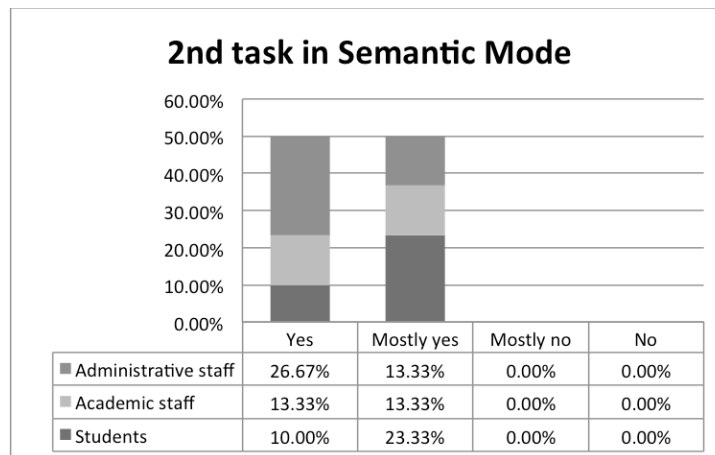
In the 1st task (see Figure 7.4.2.3 graph (a)) the vast majority of all participants responded with “Yes” (76.67% in total and almost a half – 36.67% – in the administrative staff user group), whereas the remaining 23.33% responded with “Mostly yes” (more than a half – 13.33% – in the academic staff user group). This is the best result of all three tasks. In the 2nd task (see Figure 7.4.2.3 graph (b)) both “Yes” (26.67% in the administrative staff user group) and “Mostly yes” (23.33% in student user group) have an equal number of responses in total – both 50%. This can be explained by the fact that in the 2nd task each user had to set DOI values for the selected terms. As those DOI values were assigned subjectively and each user had his/her own priority scale, it happened that some reports failed to be included into the recommendation list in semantic mode, because several DOI values were too low. In the 3rd task (see Figure 7.4.2.3 graph (c)) the largest group of participants responded with “Mostly yes” (66.66% in total and 33.33% in the student user group) and only 13.33% in the administrative staff group responded with “Yes”. Meanwhile, there were also negative responses: “Mostly no” (16.66% in total and 13.33% in the academic staff user group) and “No” (3.33% in academic staff user group). Recommendations in user activity mode are produced on the basis of all user activity during the single session (including the test task, the 1st and the 2nd task). The fact that during the session a user might have executed reports that were unnecessary for completing the preceding tasks could have had a strong influence on the resulting recommendation list in the user activity mode. However, the author should emphasize that some users commented on the recommendation list they received and stated that it conformed with the reports they executed the most during the session, even though several recommendation from that list were not helpful.

In your opinion, did the report recommendations help you complete the 1st/ 2nd/ 3rd task?

a)



b)



c)

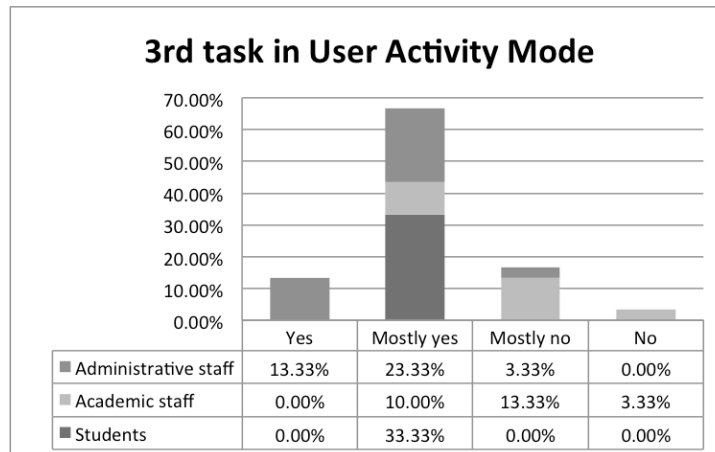


Fig. 7.4.2.3. Response results to survey questions 3, 7, and 11

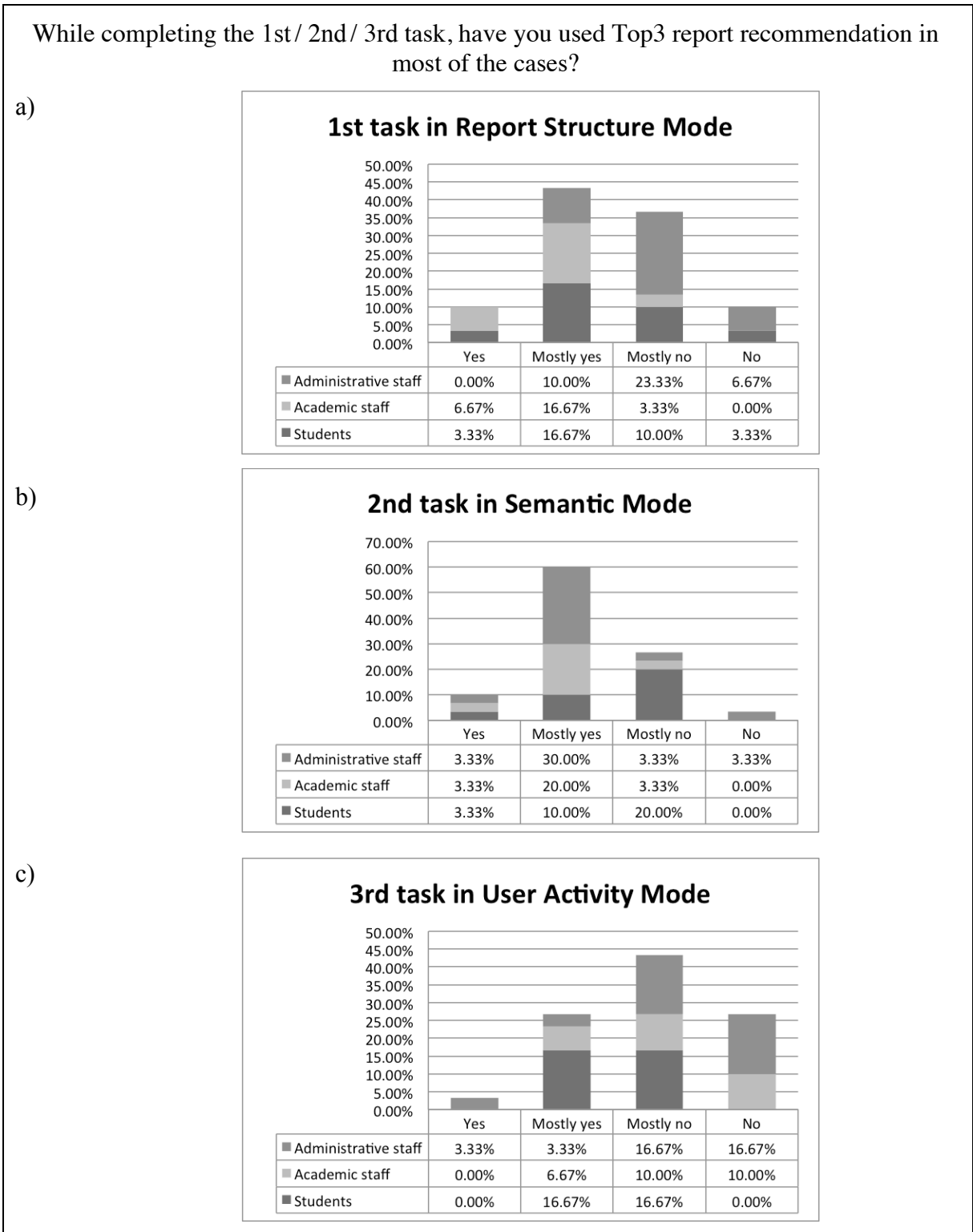
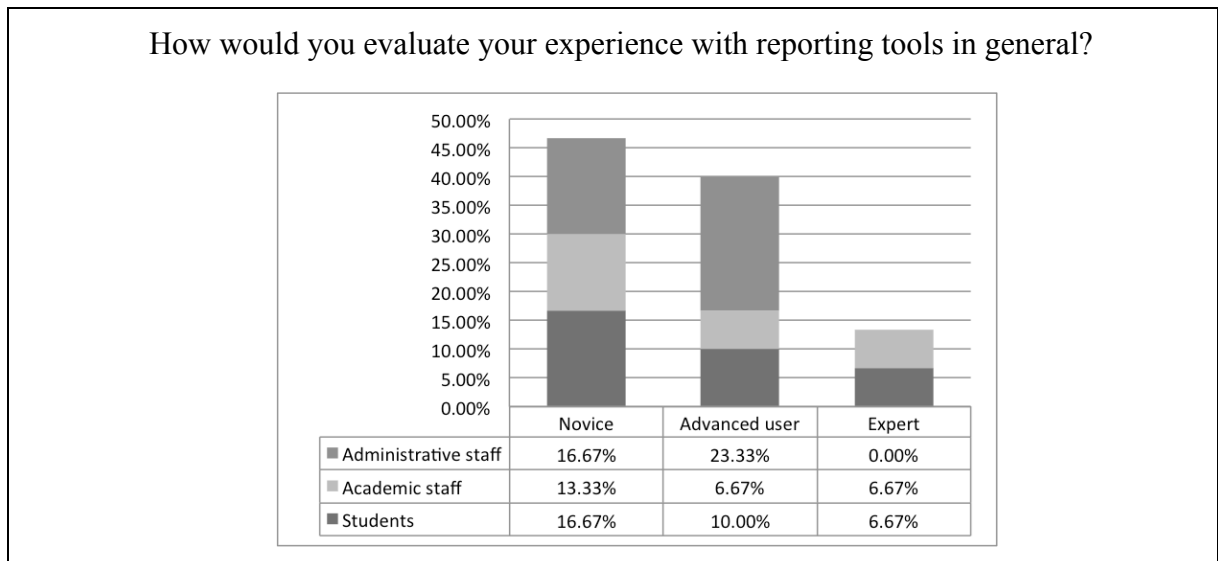


Fig. 7.4.2.4. Response results to survey questions 4, 8, and 12

In the 1st task (see Figure 7.4.2.4 graph (a)) “Mostly yes” rating value prevails (43.34% in total), followed by “Mostly no” (36.66% in total), and “Yes” and “No” rating values (10% each). Let’s see what caused high “Mostly yes” and “Mostly no” rating values. Reports that contain data required in subtasks of the 1st task don’t necessarily have an equal



or very similar structure, because all tasks in this experimentation are exploratory and are aimed at examining reports from different areas. Moreover, the quantity of workbooks that consist of 3 or more reports each is around 68% in the administrative staff user group and around 63% in the student and academic staff user groups. It means that in some cases the Top3 reports in report structure mode will be the ones from the same workbook. Taking that into consideration, the obtained rating values are satisfying. In the 2nd task (see Figure 7.4.2.4 graph (b)) “Mostly yes” rating value dominates over other values (60% in total in all user groups), because recommended reports not obligatory should have a similar structure in semantic mode. Here a similarity value is affected by the elements in the user profile, this way, a recommendation component may contain a set of rather distinct reports, which do comply with user preferences. In the 3rd task (see Figure 7.4.2.4 graph (c)) “Mostly no” rating value leads with 43.37% in total, whereas “Mostly yes” and “No” rating values follow with 26.67% each. An explanation for low results is that not all (and in some cases none) of the reports could have been found in the recommendation list in the user activity mode. This was caused by an unpredictable user activity and exploration of the reports that didn’t help in completing preceding tasks that directly influenced the recommendation.



**Fig. 7.4.2.5.** Response results to survey question 13

A graph that demonstrates how users subjectively evaluate their experience with the reporting tools in general is presented in Figure 7.4.2.5. As it is seen from the graph, user division into groups according to their skill level is true-to-life: a little less than a half of all participants (46.67%) are novice users that represent all three user groups, 40% of all participants belong to the advanced user group with the majority (23.33%) in the administrative user group, and the remaining 13.34% are expert users from academic staff and

student user group (6.67% each). Also, it demonstrates that subjects of different skill level and background have successfully mastered the reporting tool and recommendation component.

A graph in Figure 7.4.2.6 clarifies whether it was easier for users to complete the tasks employing any of the recommendation modes than to complete the task without it. There are only two response values provided by users with slight differences in percentage in each group: “Yes” (53.33% in total) and “Mostly yes” (46.67% in total). Both response values show that recommendations introduced into the reporting tool improve user experience with it, and that to a variable degree all three recommendation modes help users solve exploratory tasks. Such a high “Mostly yes” rating value can be dictated by several flaws in recommendations generated in semantic and user activity modes (see an explanation for graphs (b) and (c) in Figure 7.4.2.3).

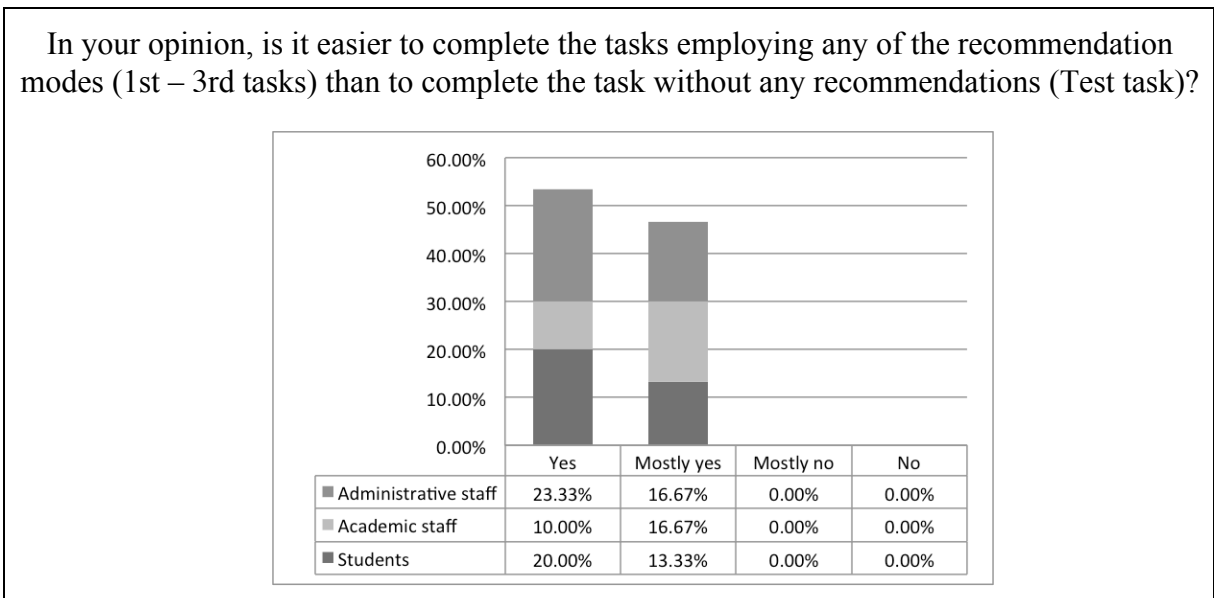


Fig. 7.4.2.6. Response results to survey question 14

A graph in Figure 7.4.2.7 demonstrates the tasks while completing which participants used recommendations more often. A graph in Figure 7.4.2.8 shows the tasks during which participants received the most precise recommendations given that a *precise recommendation* means that when a user hit a link in the recommendation component and executed the recommended report, he/she could find the necessary data required in the current subtask.

Although there is some noticeable correlation between graphs in Figure 7.4.2.7 and Figure 7.4.2.8, let’s examine the difference between these two graphs. When subjects stated 2 tasks (i.e. 1st and 2nd, and 2nd and 3rd) in Figure 7.4.2.7 where they had used recommendations most of all, some of them selected only one task during which they had

received the most precise recommendations. As the graph in Figure 7.4.2.8 shows, participants voted mostly in favor of the 2nd task in semantic mode (especially, student and administrative staff user groups), whereas a significant part of academic staff user group preferred the 1st task. Thus, the survey shows that in the aspect of recommendation precision the 2nd task in semantic mode prevails over others with a rating value of 50% in all user groups in total.

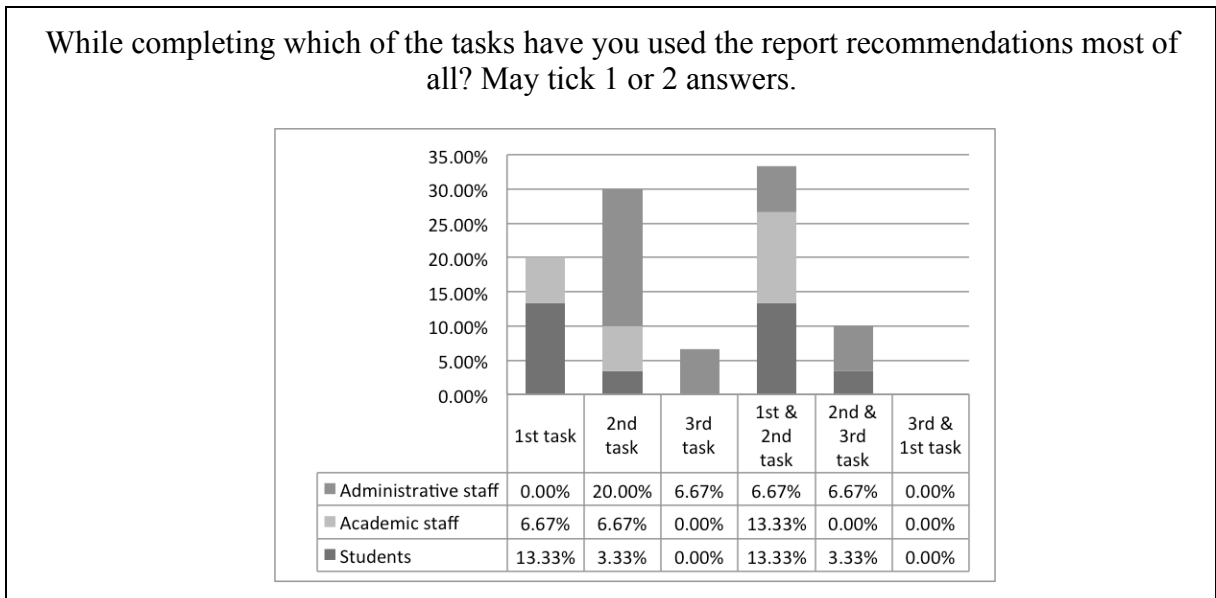


Fig. 7.4.2.7. Response results to survey question 15

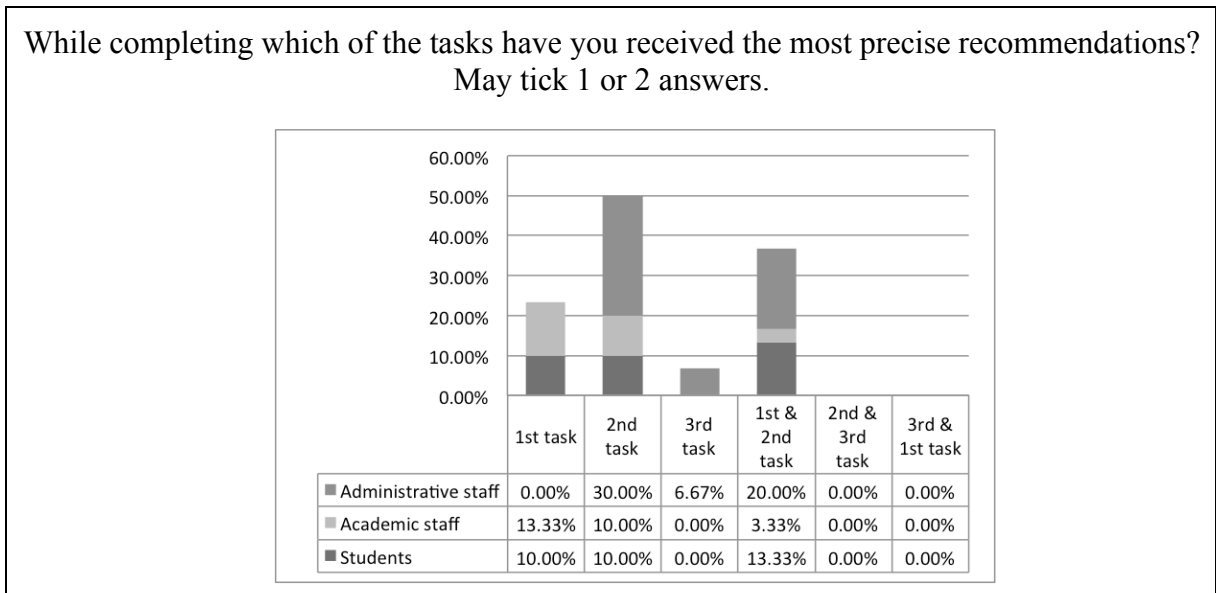


Fig. 7.4.2.8. Response results to survey question 16

### 7.4.3. Reporting Results of the User Feedback

Leaving a comment or a suggestion after filling in the survey was not a mandatory task, however, the majority of the subjects – 25 out of 30 – gladly provided their feedback. All comments have been given in a free form, which is why the author had to classify and sort them on her own. There are two groups of feedback included in this thesis: the one that gives a subjective rating to report execution in recommendation modes and the other feedback, which either includes ideas on what to improve in user interface/functionality of the reporting tool and its recommendation component or overall impressions/concerns.

**Table 22. Subject comments on recommendation modes from user survey and their preferred mode(s)**

<b>Subject Feedback on Recommendation Modes</b>	<b>Which recommendation mode is the most preferred one?</b>		
	Report Structure	Semantic	User Activity
“User activity mode could be the most frequently used in everyday life, because of the accumulating user activity history. Thus, most used or similar reports would be right at hand. For an inexperienced user report structure mode is the best, because it compares reports dynamically and reports of interest are easy to find checking the similarity coefficient values.”	1		1
“Semantic mode seemed to be the most effective, because there a user could state preferences.”		1	
“Report structure mode was the best mode to work in. Semantic mode seemed good too, although it required some extra effort from a user. Recommendations in user activity mode didn't help to complete the task. It seemed that most likely it was because a lot of different (partially irrelevant) reports were looked through during the session.”	1		
“Semantic mode performed best of all, because it was possible to find easily all necessary report recommendations to complete the task. During the session there have been some reports that should not have been executed, which affected recommendations in user activity mode.”		1	
“Report structure is the best mode to work in. It is also good to work in semantic mode, if a user doesn't have to define too many preferences (say, up to 5), otherwise it is hard to interpret recommendations and a user has to keep in mind all the preferred elements. This mode would be the most effective, if a user knew the terms of interest beforehand.”	1		
“Semantic mode could be the most frequently used, because here it is possible to precisely define elements of interest. User activity mode is useful, if one has to execute reports on a regular basis. Report structure mode seems to be the least used mode.”		1	
“The most convenient mode in usage is semantic mode.”		1	
“The best modes are report structure and user activity, because here a user does not have to study in depth how to state user preferences. A possible working scenario might be that a user employs recommendation modes that do not require user assistance first, and then switches to semantic mode to define preferences, if it is necessary.”	1		1
“Surely recommendation component does its job and produces valid recommendations. Moreover, recommendations differ in each of the	1	1	1

<p>modes. For instance, in report structure mode recommended reports indeed had similar structure. In semantic mode recommendations conformed to preferences stated in user profile. In user activity mode recommended reports were indeed similar to the most used ones.”</p>			
<p>“The best modes are report structure and user activity modes, however, the best results were in user activity mode, because the recommendation component produced very precise recommendations that were easy to focus on. The most sophisticated mode was semantic mode, because here a user has to define preferences and DOIs explicitly.”</p>			1
<p>“The best mode is semantic mode, because here a user is capable of affecting recommendations.”</p>		1	
<p>“A user is capable of formulating his/her preferences in semantic mode, if he/she is well-informed on reports that he/she wants to find. If a user lacks information or experience, then it is easier to work in report structure and user activity modes. If everyday tasks are monotonous, then the most suitable mode is user activity mode.”</p>			1
<p>“In general, working with this reporting tool in different recommendation modes was a good experience. Semantic mode, where a user may influence recommendations, made the best impression. It seems that if a user knows his/her interests, then this mode will be very helpful. If a user does not execute reports often (for example, one in 6 months), then in this case a user might like user activity mode, which "remembers" reports of interest instead of the user.”</p>		1	
<p>“Recommendation modes help in finding necessary reports. One has to switch between recommendation modes for a better effect. All in all, an advanced user that is aware of his/her interests would employ semantic mode, however, report structure mode would come in handy for a user who has to find certain data. User activity mode turns out to be the least effective, because not always reports that a user has executed, say, out of curiosity are the ones that really interest him/her.”</p>	1	1	
<p>“Report structure mode might be the one to work best for inexperienced users.”</p>	1		
<p>“It is hard to evaluate user activity mode in such a short time, although it showed executed or similar reports as recommendations. Both report structure and semantic modes work well, but in semantic mode a lot depends on DOI values that a user enters. It can happen that DOI value is too low, which is why some reports do not appear in recommendation list.”</p>	1		
<p>“Semantic mode allows getting the most predictable results, because the order of recommended reports fully corresponded to DOIs assigned to user-selected terms. First, a user should learn how to work with the reporting tool, and then it user activity mode would come in handy, because it is suitable for routine tasks bearing in mind that a user does not have to memorize any of his/her most frequently used reports. Also, a benefit of user activity mode is that one doesn't have to state any user preferences.”</p>		1	
<p>“All three recommendation modes seem to be effective. In each mode it was not possible to find the required report by means of recommendations only once. If there were a task to complete with this reporting tool in real life, then the most preferred modes would be report structure and semantic mode. It was also easy to work with the reporting tool without recommendations, because of the good knowledge of this information domain.”</p>	1	1	
<p>“The most valuable mode is semantic mode, because it allows to adjust report recommendations to some specific needs of a user, thus, this mode is the most flexible.”</p>		1	
	<b>9</b>	<b>11</b>	<b>5</b>

Table 22 lists all subject feedback that evaluates their experience with recommendation modes in the reporting tool. Here the subject feedback is presented as it was left in each survey. The author has rated each comment with either "1" or null in each recommendation mode to summarize in a compact way the most preferred recommendation mode(s). As one may conclude from the totals given in the bottom of the Table 22, semantic mode (total = 11) is favored over report structure (total = 9) and user activity (total = 5) modes. Now, let's summarize subject feedback from the Table 22.

*Report structure mode* works best for an inexperienced user who can explore the reporting tool by means of recommendations. The benefit of this mode is that it doesn't require user assistance. As some participants of the experimentation noticed, recommended reports indeed had a similar structure.

*Semantic mode* seemed to be the most effective and flexible, because there a user could precisely state preferences, although it required some extra effort from a user and for that reason seems more sophisticated. It would be most suitable for an advanced user who could formulate his/her interests and would perform best, if a user didn't have to define too many preferences (say, not more than 5), otherwise it would have been hard to interpret recommendations. Some subjects noticed that the order of recommended reports fully corresponded to DOIs assigned to user-selected terms. All three recommendation modes seem to be effective and produce realistic recommendation lists. A possible working scenario offered by an experimentation participant might be that a user employs recommendation modes that do not require user assistance first, and then switches to semantic mode to define preferences, if it is necessary.

*User activity mode* is the one that is hard to evaluate in such a short time (1 session), although it could be the most frequently used mode in everyday life to complete monotonous tasks or execute reports on a regular basis because of the accumulating user activity history. For some subjects recommendations in user activity mode didn't help good enough to complete the task, because a lot of reports were looked through during the session out of curiosity, although recommended reports were indeed similar to the most often-used ones. In its turn, for those subjects who did not create any irrelevant activity this mode produced very precise recommendations that were easy to focus on. Just like in report structure mode, the benefit of this mode is that it doesn't require user assistance. Another advantage of this mode could be the ability to "remember" reports of interest instead of the user, which is especially valuable, if one doesn't execute reports often (for example, once in 6 months).

Another part of subject feedback includes rather unstructured, however, still valuable suggestions. For example, ideas on user interface improvement (which have been considered as valid and implemented in the reporting tool) are the following:

- “It would be good to have a preview of a report before execution, so that a user could see table columns (or rows and columns of a crosstab) and decide whether this report contains elements of interest or not”;
- “Probably the last report execution date by the user could be shown in recommendations, thus, providing more information on recommended reports”;
- “A button to show/hide a recommendation component could be added”;
- “Time interval for user activity analysis in user activity mode might be extended to 1 year, because usually in enterprises such time intervals are used”.

Some more interesting suggestions:

- “In user activity mode it would have been interesting to see recommendations of other users, in case if duties and user rights in the reporting tool coincide”;
- “Recommendations did not help in finding the initial report in each task. It would have been better, if recommendations appeared not after execution of the initial report, but earlier (for example, at the home screen)” – this idea can be implemented only in semantic and user activity modes;
- “It could have been easier to choose elements for user preferences in semantic mode in a interactive way, for instance, using a "tree" structure to select terms”.

Also, some subjects left their general impressions:

- “The reporting tool itself is well-structured and intuitively understandable”;
- “Recommendations in the reporting tool make it easy-to-use and considerably speed up the process of report searching”;
- “For someone who doesn't have any knowledge in this information domain it is hard to interpret the reports”;
- “In general, working with this reporting tool in different recommendation modes was a good experience”;

In addition, there were several concerns in a form of an open question:

- “How would recommendation modes perform in real life?”

In author’s opinion, to answer this question, a long-term usage of the reporting tool with the recommendation component is needed with the succeeding description of the real-life exploratory tasks, assessment of recommendation quality and usefulness provided by the user. A *long-term* means at least 12 months, because, in case of the

data mart employed for the experimentation, the time period when the data in reports reflects updates may vary (say, typical changes in the number of registered students, their final grades, etc. occur once per semester, which is why certain reports don't require being executed more often);

- “Recommendations mostly are a good help in completing tasks, but the question is how much does a user trust this system in general?”

From the author's point of view, trust correlates with the quality of data in reports. Technically, random data checks for its validity may help to decide on how much one trusts the system in general. Also, trust to the generated recommendations can increase (or decrease) during the long-term usage of the reporting tool with and without recommendations. Thus, a user is able to estimate, whether the recommendations help in solving regular tasks, finding the necessary data, and improving the overall experience with the reporting tool.

## **7.5. Summary of the Section**

There is much controversy in explicit methods for gathering user preferences [GSCM07], 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.

All in all, even though semantic mode is the one where a user has to do some extra work by stating his/her preferences explicitly and the task in this mode was mostly qualified as “Average” (while other tasks seemed “Easy”), it was the most preferred mode in subject feedback. Moreover, the ability to affect and control recommendations is mostly considered as an advantage. Log-table analysis showed that there is no significant difference in performance of the recommendation component in report structure and semantic modes, however, in report structure or in semantic mode the recommendation component outperforms that in user activity mode. There is a marginally significant difference in the performance of generating recommendations between modes that gather user preferences implicitly and the one that gathers it explicitly in favor of the latter.

Also, survey results showed that experimentation participants considered that the most precise recommendations were produced in semantic mode. As to the modes where recommendations are generated on the basis of implicitly stated user preferences (which was also appreciated by users), report structure mode is a “runner-up”, while user activity mode stays a little underrated. Subjects confirmed the initial thoughts of the author by stating that



report structure mode would perform best for users who lack experience in the reporting tool. As some subjects notice, user activity mode would have more value in the long run and would suit best for users who have to execute a set of reports on a regular basis.

User surveys results were also split into two groups according to user experience with reporting tools – i.e. novice (inexperienced users) vs. advanced users and experts (experienced users). The results are represented as a table in Appendix 6. In the estimation of most participants in both user groups the most complex task was in semantic mode (rated as “Average”), and qualified as “Mostly clear”. However, in spite of it, an overwhelming majority in both user groups regardless of their experience stated that the most precise recommendations were received in semantic mode. Recommendations in all three modes helped (i.e. “Yes”, ”Mostly yes”) subjects of both groups to complete the tasks, although, the task in user activity mode was the only one that had also negative responses (i.e. “Mostly no” – in both user groups, ”No” – in experienced user group). This may be explained by the fact that experienced users work with the reporting tool with more confidence, explore and execute the larger number of reports including the irrelevant ones. This way, their activity history is richer and contains reports that should not have been executed in all of the previous tasks, thus, leading to erroneous recommendations.

In general, one may conclude that user activity mode shows comparatively worse results in terms of one session irrespective of the experience of the user. Subjects of experienced user group claimed that they used recommendation component most of the time in semantic mode, meanwhile, novice users showed preference for both report structure and semantic mode. This leads to a conclusion that semantic mode, which requires extra effort in defining user preferences, is suitable for experienced users, whereas novice users prefer either structure mode as an implicit way of stating preferences or semantic mode as an explicit one.

## 8. CONCLUSIONS

The field of personalization in OLAP still is being explored among the researchers worldwide. As stated in [GR09a], personalization in data warehouses still deserves more attention by researchers and needs to be examined more thoroughly both on theoretical and practical level, despite numerous studies initiated by [Kie02, Cho02] and continued by other researchers on user preferences in the field of databases. There are three main reasons to study personalization in data warehouses [GR09a]: (i) user preferences allow a user to focus on the data that seems to be the most essential, more precisely – while composing and executing queries, user preferences would be a natural way how to avoid both an empty set of results and data flooding; (ii) preferences allow user to specify a pattern of what data to select as during OLAP sessions a user might not know exactly what kind of data he/she is looking for; and (iii), it would be worthwhile to give a user an opportunity to express preferences on aggregated data.

A motivation for the author to work on thesis about OLAP personalization was the experience in using standard commercial applications for producing and managing data warehouse reports (for instance, Oracle Business Intelligence Discoverer<sup>8</sup> and MicroStrategy<sup>9</sup>) at the University of Latvia as well as participation in scientific projects and development of a new data warehouse reporting tool [Sol07]. The data warehouse reporting tool was chosen as a suitable environment for implementing and testing the developed techniques of OLAP personalization.

### 8.1. Results of the Research

The results acquired in the course of the research are the following:

- A subject for a new study was defined as generating recommendations in a data warehouse reporting tool on the basis of user preferences on logical metadata (OLAP schema, its elements, and aggregate functions). A comparative analysis of the state-of-the-art approaches was performed in order to categorize and characterize them, and to identify a gap in research and which of the approaches would be the most suitable for a new empirical study in the area of the data warehouse personalization. It was decided that (i) user preferences could be stated either implicitly or explicitly and interpreted as soft constraints, (ii) business terms would be used for formulating user

---

<sup>8</sup> Oracle Business Intelligence Discoverer available at: <http://www.oracle.com/technetwork/developer-tools/discoverer/overview/index.html>

<sup>9</sup> MicroStrategy available at: <http://www.microstrategy.com/software/products/report-services>

preferences in a way that is more understandable for a user, since this aspect wasn't discussed in any of the reviewed approaches;

- The extended requirement formalization metamodel was developed, which serves to minimize the risk while defining and processing information requirements and to ensure that the succeeding construction of the conceptual model of a data warehouse is aligned with user needs stated as information requirements. The risk of interpreting information requirements erroneously is threefold: a client might be imprecise in formulating the needs, an interviewer might capture them incorrectly, and, finally, a developer might construct a conceptual model that does not fully comply with information requirements stated by the client. According to [AG12], user requirements are a personalization factor, which is not fully exploited in data warehouses;
- A method has been proposed, which provides an exhaustive description of interaction between a user and a data warehouse using the concept of Zachman Framework [Zac, Zac03]. In accordance with this framework a composite user profile consisting of a set of generic user-describing profiles (user, interaction, temporal, spatial, preferential, and recommendational) has been developed. In this thesis special attention was paid to suggesting possible recommendations for novice and experienced users of the new OLAP reporting tool based on their preferences collected in preferential profiles;
- OLAP preferences metamodel, which serves for formulating user preferences for OLAP schema elements and aggregate functions, has been proposed based on the empirical studies of reporting tools. Since OLAP preferences metadata is compatible with other metadata layers (i.e. logical, physical, report, and semantic) of the OLAP reporting tool [Sol08a], OLAP preferences metadata got integrated with all other metadata layers;
- Three distinct content-based methods for construction of report recommendations have been developed: *hot-start* method that takes advantage of the user activity log, *cold-start* method that defines similarity of reports based on their structure, and *semantic hot-start* method that employs user-defined preferences for report elements. The methods distinguish and recommend reports that potentially may interest the user taking advantage of user preferences for data warehouse schema elements and aggregate functions. The recommendation component based on these methods was integrated into the OLAP reporting tool.
- The experimental study was performed in laboratory settings involving 30 subjects with various level of experience with reporting tools (novice/advanced user/expert). The aim of the experimentation was to explore which of the methods for generating

recommendations in the reporting tool would produce more accurate recommendations. A data mart to gather data on user interaction with Moodle course management system (referred as Moodle or Moodle CMS) and study process in the University of Latvia was designed and developed. An experiment was conducted with a maximum number of 70 reports accessible for each group of subjects. To evaluate each method and compare with others, user activity log was analyzed as well as direct feedback on the methods was gathered in a form of user survey and processed.

The results of the research study conducted in terms of this thesis on the subject of personalization in data warehouses were presented at international scientific conferences such as BIR (Perspectives in Business Informatics Research), ISD (Information Systems Development), and ICEIS (International Conference on Enterprise Information Systems) and published. Altogether 8 scientific papers [KN10, KN11, KS11, SK11, NNK11, KS12, Koz13, KN14] were produced by the author and served as a contribution to this thesis. The most cited paper (currently having 13 citations in Google Scholar and 6 citations in Scopus) is [KN10], which is referred to by other researchers in their latest papers, for example, [AG12, BK13, KB13, BRBC14]. Apart from that, the results of the thesis were presented at local scientific conferences such as Scientific Conference of the University of Latvia in 2010, 2011, and 2013, and a conference in terms of ESF project in 2011 and 2012.

## **8.2. Conclusions on the Experimental Study**

All three methods – hot-start, cold-start, and semantic hot-start – were implemented in the recommendation component of the OLAP reporting tool in user activity, report structure, and semantic modes respectively and approbated in terms of the experimental study. The aim of the experimentation was to explore which of the implemented methods for generating recommendations and which type of gathering user preferences (implicit or explicit) produces more accurate recommendations that lead to completing the task using the recommendation component of the reporting tool extensively. There were 30 subjects who took part in the experimentation grouped by their rights on reports (student/academic staff/administrative staff) and skill level (novice/advanced user/expert).

Analysis of the results of the experimental study was threefold and results were gathered from such sources as: log-table, user survey, and user comments given in a free form.

Log-table analysis showed that there is no significant difference in performance of the recommendation component in report structure and semantic modes, however, in report

structure or in semantic mode the recommendation component outperforms that in user activity mode.

User survey results showed that experimentation participants considered that the most precise recommendations were produced in semantic mode (regardless of their skill level).

Summary of the user feedback helped to conclude that semantic mode is more suitable for experienced users, whereas novice users prefer either structure mode as an implicit way of stating preferences or semantic mode as an explicit one; subjects found it hard to evaluate the user activity mode in just one session time, although it could be the most frequently used mode in everyday life to complete monotonous tasks.

Considering the type of gathering user preferences, log-table analysis showed that there is a marginally significant difference in the performance of generating recommendations between modes that gather user preferences implicitly (i.e. report structure and user activity modes) and the one that gathers it explicitly (semantic mode) in favor of the latter. In addition, user feedback revealed that even though the preferences in semantic mode are stated explicitly that requires an extra effort, this mode is the most preferred one comparing to others.

### **8.3. Conclusions on the Research Goal and Formulated Hypotheses**

The goal of this doctoral thesis was to provide new methods to support personalization in the OLAP reporting tool delivering data that satisfies user needs. A set of corresponding tasks has been fulfilled successfully to reach this goal. The new methods suitable for novice, advanced, and expert users were empirically tested in terms of the experimentation and approved by the participants of the experimental study.

There were two hypotheses stated in the beginning of the research. Let's consider whether each of them got approved or rejected.

The 1st hypothesis: "Integration of personalization into the data warehouse reporting tool can save effort of the user during the working sessions with the reporting tool". This hypothesis is approved by results of the experimental study with the recommendation component in the OLAP reporting tool. In terms of the experimentation, the subjects completed one exploratory task without any report recommendations, and the remaining three – applying each of the proposed methods, namely, cold-start, semantic hot-start, and hot-start implemented in the recommendation component. The results of the experimental study showed that all of the methods for generation of report recommendations were positively evaluated in terms of saving user effort. The participants were asked to compare, whether it was easier to complete the tasks with the help of report recommendations than without them;

53.33% of all respondents answered “Yes” and the remaining 46.67% replied with “Mostly yes”.

The 2nd hypothesis: “Methods for generation of recommendations in OLAP that take as input user preferences gathered implicitly or explicitly and are suitable for different groups of users may be proposed”. This hypothesis is approved by the results directly gathered from user surveys. Summary of the user feedback helped to conclude that semantic hot-start method (semantic mode) is more suitable for experienced users, whereas novice users prefer either cold-start method (structure mode) as an implicit way of stating preferences or semantic hot-start method (semantic mode) as an explicit one; subjects found it hard to evaluate the hot-start method (user activity mode) in just one session time, although it could be the most frequently used method in everyday life, which would help in completing monotonous tasks.

#### **8.4. Discussions and Limitations on the Research**

There are certain limitations for application of the methods for generation of report recommendations:

- The methods proposed by the author of the thesis that employ OLAP preferences to generate report recommendations, namely, cold-start, hot-start, and semantic hot-start methods, exploit schema-specific OLAP preferences only (see section 5.2.5). It was decided to concentrate on schema-specific OLAP preferences, due to the lack of research results by other authors on the methods for generating recommendations on the basis of OLAP schema elements;
- Recommendations in the reporting tool are generated individually for each user taking as an input his/her preferences only. It is done this way, because users of the reporting tool might have different rights on reports. Thus, recommendations generated for a group of users with similar preferences, might be of little help to a certain user, because he/she doesn't have the rights to execute a number of report(s) from the recommendation list.

The OLAP reporting tool needs to be further developed in terms of the technical implementation, namely, in the aspect of usability, as concluded from user feedback. Besides, it would be beneficial to involve some users into exploiting the reporting tool with the recommendation component for a long period of time on a regular basis. The feedback that such a user would give, could be compared with the results acquired in the process of the experimental study presented in this thesis.

Certain improvements in all three methods for generation of report recommendations may be considered such as, for example, collecting user feedback on received report recommendations (i.e. a “yes/no” answer to the question “was the recommendation helpful?”). This feedback might be integrated into the calculation of similarity values in each of three proposed methods for generation of report recommendations, thereby, allowing users to interactively state their opinion on the received recommendations and improve the quality of the future ones.

Other direction is the development of the technical application of the recommendation component. There may be considered an idea of making the recommendation component a parametrized module that would be compatible not only with the OLAP reporting tool developed in the University of Latvia, but also with other reporting tools, physical, logical, and semantic metadata of which support CWM standard [CWM].

Personalization in OLAP is a relatively new and developing field of research that keeps on being widened with new approaches. The author is convinced that the OLAP personalization approach consisting of three distinct methods described in this thesis occupies its own niche in report recommendations. Nevertheless, there are new perspectives of OLAP personalization to be considered. For instance, an approach that would handle a complete personalization process offering a multi-faceted personalization (i.e. starting with the processing of user requirements and finishing with visualization of reports) is yet to be developed.

## **ACKNOWLEDGMENTS**

The author would like to thank Dr. sc. comp. Laila Niedrīte, the advisor of this doctoral thesis, for her constructive remarks and assistance, all participants of the experimentation for their responsiveness, patience, support and willingness to contribute to this study, and family for their understanding during the whole process of thesis writing.

This doctoral thesis has been supported by the European Social Fund within the project “Support for Doctoral Studies at University of Latvia”.



## REFERENCES

- [AG12] Aissi, S., Gouider, M.S. 'Towards the Next Generation of Data Warehouse Personalization System: A Survey and a Comparative Study'. *International Journal of Computer Science Issues (IJCSI)*, 2012, 9(3-2):561-568.
- [AMK11] Adomavicius, G., Manouselis, N., Kwon, Y.-O. 'Multi-Criteria Recommender Systems'. In: Ricci, F., et al. (eds) *Recommender Systems Handbook*, Springer, Springer Science+Business Media LLC, 2011, Part 5, pp. 769-803.
- [AW00] Agrawal, R., Wimmers, E. 'A Framework for Expressing and Combining Preferences'. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, New York, 2000, pp. 297-306.
- [Bas92] Basili, V. 'Software Modeling and Measurement: The Goal/Question/Metric Paradigm'. *Computer Science Technical Report Series, CS-TR(2956)*, University of Maryland, 1992, 24 pages.
- [BD] Business Dictionary: Definition of the Personalization [online] <http://www.businessdictionary.com/definition/personalization.html>
- [BGMM06] Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H. 'Personalization of MDX Queries'. In *Proceedings of XXIIemes journees Bases de Donnees Avancees (BDA'06)*, Lille, France, 2006.
- [BHS+98] Ballard, C., Herreman, D., Schau, D., et. al. 'Data Modelling Techniques for Data Warehousing'. Sommers (NY), IBM Corporation, 1998, 216 p.
- [BHK98] Breese, J.S., Heckerman, D., Kadie, C. 'Empirical Analysis of Predictive Algorithms for Collaborative Filtering'. In: *Proceeding of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98)*, Madison, WI, USA, 1998, pp. 43-52.
- [BK13] Bentayeb, F., Khemiri, R. 'Adapting OLAP Analysis to User's Constraints through Semantic Hierarchies'. In *Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS'13)*, Angers, France, 2013, vol. 1, pp. 160-167.
- [BKS01] Borzsonyi, S., Kossmann, D., Stocker, D. 'The Skyline Operator'. In *Proceedings of the 17th International Conference on Data Engineering*, Heidelberg, April, 2001.
- [BRBC14] Belo, O., Rodrigues, P., Barros, R., Correia, H. 'Restructuring Dynamically Analytical Dashboards Based on Usage Profiles'. In *Proceedings of the 21st International Symposium on Methodologies for Intelligent Systems (ISMIS'14)*, Roskilde, Denmark. Springer Verlag, 2014, LNAI, vol. 8502, pp. 445-455.
- [Bur02] Burke, R. 'Hybrid Recommender Systems: Survey and Experiments'. Kluwer Academic Publishers, USA, 2002, 12(4):331-370.
- [Bur07] Burke, R. 'Hybrid Web Recommender Systems'. *The Adaptive Web – LNCS*, vol. 4321, Springer, Heidelberg, 2007, pp. 377-408.

- [B09] A Guide to the Business Analysis Body of Knowledge<sup>®</sup> (BABOK<sup>®</sup> Guide), version 2. International Institute of Business Analysis, Toronto, Ontario, Canada, 2009.
- [CG13] Chaibi, N., Gouider, M.S. 'Personalization and Recommendation of Queries in Multidimensional Data Base'. *International Journal of Engineering Science Invention (IJESI)*, 2013, 2(5):74-80.
- [Cho02] Chomicki, J. 'Querying with intrinsic preferences'. In *Proceedings of the 8th International Conference on Advances in Database Technology (EDBT)*, Prague, Czech Republic, 2002, pp. 34-51.
- [Cho03] Chomicki, J. 'Preference Formulas in Relational Queries'. *ACM TODS*, 28(4), 2003, pp. 427-466.
- [CWM] Object Management Group: Common Warehouse Metamodel Specification, v1.1, [online] <http://www.omg.org/cgi-bin/doc?formal/03-03-02>
- [DHK08] Drachsler, H., Hummel, H. G. K., Koper, R. 'Personal Recommender Systems for Learners in Lifelong Learning Networks: the Requirements, Techniques and Model'. *International Journal of Learning Technology*, 2008, 3(4):404-423.
- [ESSD08] Easterbrook, S., Singer, J., Storey, M.A., Damian, D. 'Selecting Empirical Methods for Software Engineering Research'. Shull, F. et al. (eds.): *Guide to Advanced Empirical Software Engineering*. Springer-Verlag, London, 2008, pp. 285-311.
- [FHKS08] Frank, U., Heise, D., Kattenstroth, H., Schauer, H. 'Designing and Utilizing Business Indicator Systems within Enterprise Models - Outline of a Method'. In: Loos, P., et al. (eds.) *Modellierung betrieblicher Informationssysteme (MobIS'08) - Modellierung zwischen SOA und Compliance Management*, Saarbrücken, Germany, LNI, vol. 141, 2008, pp. 89-105.
- [FIA] Find IP Address: IP Lookup. [online] <http://www.find-ip-address.org/>
- [GBC+06] García, F., Bertoa, M., Calero, C., et. al. 'Towards a Consistent Terminology for Software Measurement'. *Information and Software Technology*, 2006, 48(8):631-644.
- [GBR07] Guédria, W., Bellahsene, Z., Roche, M. '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, 2007, pp. 21-26.
- [GG06] Garrigós, I., Gómez, J. 'Modeling User Behaviour Aware WebSites with PRML'. In *Proceedings of the CAiSE'06 Third International Workshop on Web Information Systems Modeling (WISM'06)*, Luxemburg, June 5-9, 2006, pp. 1087-1101.
- [GMNS09] Giacometti, A., Marcel, P., Negre, E., Soulet, A. 'Query Recommendations for OLAP Discovery-driven Analysis'. In *Proceedings of 12th ACM International Workshop on Data Warehousing and OLAP (DOLAP'09)*, Hong Kong, November 6, 2009, pp. 81-88.

- [GMNS11] Giacometti, A., Marcel, P., Negre, E., Soulet, A. 'Query Recommendations for OLAP Discovery-driven Analysis'. *Data Warehouse Mining*, 2011, 7(2):1-25.
- [GMR98] Golfarelli, M., Maio, D., Rizzi, S. 'Conceptual Design of Data Warehouses from E/R schemes'. In *Proceedings of 31st Annual Hawaii International Conference on System Sciences (HICSS'98)*, Kona, Hawaii, IEEE, USA, vol. 7, 1998, pp. 334-343.
- [GPMT09] Garrigós, I., Pardillo, J., Mazón, J.-N., Trujillo, J. 'A Conceptual Modeling Approach for OLAP Personalization'. *Conceptual Modeling - ER 2009*, LNCS, vol. 5829, Springer, Heidelberg, 2009, pp. 401-414.
- [GR09a] Golfarelli, M., Rizzi, S. 'Expressing OLAP Preferences'. Berlin / Heidelberg, LNCS, vol. 5566/2009, *Scientific and Statistical Database Management*, 2009, pp. 83-91.
- [GR09b] Golfarelli, M., Rizzi, S. 'Data Warehouse Design: Modern Principles and Methodologies'. McGraw-Hill, 2009, 480p.
- [GRG08] Giorgini, P., Rizzi, S., Garzetti, M. 'A Goal-oriented Approach to Requirement Analysis in Data Warehouses'. *Decision Support Systems*, 2008, 45(1):4-21.
- [GSCM07] Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A. 'User Profiles for Personalized Information Access'. Brusilovsky, P. et al. (eds.): *The Adaptive Web* (chapter 2). Springer-Verlag, Berlin, Heidelberg, 2007, LNCS 4321, pp. 54-87.
- [HK05] Hafenrichter, B., Kießling, W. 'Optimization of Relational Preference Queries'. In *Proceedings of the 16th Australasian Database Conference, ADC 2005*, vol. 39, Newcastle, Australia, 2005, pp. 175-184.
- [Inm02] Inmon, W. H. 'Building the Data Warehouse, 3rd ed.' Wiley Computer Publishing, 2002, 428p.
- [IGG03] Imhoff, C., Gallemmo, N., Geiger, J. G. 'Mastering Data Warehouse Design: Relational and Dimensional Techniques'. Wiley Publishing, USA, 2003, 456p.
- [IPAG] IP Address Geolocation to Identify Website Visitor's Geographical Location. [online] <http://www.ip2location.com/>
- [ITU03] ITU-T – International Telecommunications Union: Recommendation Z.150 (02/03), User Requirements Notation (URN) – Language Requirements and Framework. Geneva, Switzerland, 2003.
- [JKP04] Jensen, C. S., Kligys, A., Pedersen T. B., Timko, I. 'Multidimensional Data Modeling for Location-based Services'. *The VLDB Journal — The International Journal on Very Large Data Bases*, 2004, 13(1):1-21.
- [JRTZ09] Jerbi, H., Ravat, F., Teste, O., Zurfluh, G. 'Preference-Based Recommendations for OLAP Analysis'. In *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'09)*, Linz, Austria, 2009, pp. 467-478.
- [JRTZ11] Jerbi, H., Ravat, F., Teste, O., Zurfluh, G. 'A Framework for OLAP Content Personalization'. In *Proceedings of the 14th east European conference on*

- Advances in databases and information systems (ADBIS'10). Springer-Verlag, Berlin, Heidelberg, 2011, LNCS, vol. 6295, pp. 262-277.
- [JS05] Jones, M. E., Song, I.-Y. 'Dimensional Modeling: Identifying, Classifying & Applying Patterns'. In: Proceedings of ACM 8th International Workshop on Data Warehousing and OLAP (DOLAP'05), Bremen, Germany, 2005, pp. 29-37.
- [Kie02] Kießling, W. 'Foundations of preferences in database systems'. In Proceedings the International Conference on Very Large Databases (VLDB'02), Hong Kong, China, 2002, pp. 311-322 .
- [Kie06] Kießling, W. 'Preference Handling in Database Systems'. Talk at L3S, University of Hannover, February 06, 2006.
- [Kim02] Kim, W. 'Personalization: Definition, Status, and Challenges ahead'. Journal of Object Technology, 2002, 1(1):29-40. [online] [http://www.jot.fm/issues/issue\\_2002\\_05/column3/](http://www.jot.fm/issues/issue_2002_05/column3/)
- [Koz10] Kozmina, N. 'Datu noliktavu personalizācija'. Master thesis, University of Latvia, 2010. [online] [http://www.lu.lv/fileadmin/user\\_upload/lu\\_portal/projekti/datorzinatnes\\_pielietojumi/zinas/atkaites/Natalija\\_Kozmina\\_magistra\\_darbs.pdf](http://www.lu.lv/fileadmin/user_upload/lu_portal/projekti/datorzinatnes_pielietojumi/zinas/atkaites/Natalija_Kozmina_magistra_darbs.pdf)
- [Koz13] Kozmina, N. 'Adding Recommendations to OLAP Reporting Tool'. In Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS'13), Angers, France, 2013, vol. 1, pp. 238-245.
- [KB12] Khemiri, R., Bentayeb, F. '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.
- [KB13] Khemiri, R., Bentayeb, F. 'User Profile-Driven Data Warehouse Summary for Adaptive OLAP Queries'. International Journal of Database Management Systems (IJDMS), 2013, 4(6):69-84.
- [KEW11] Kießling, W., Endres, M., Wenzel, F. 'The Preference SQL System - An Overview'. Bulletin of the Technical Committee on Data Engineering, IEEE CS, 2011, 34(2):11-18.
- [KI04] Koutrika, G., Ioannidis, Y. E. '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, 2004, pp. 597-608.
- [KK02] Kießling, W., Köstler, G. 'Preference SQL-Design, Implementation, Experiences'. In Proceedings of the International Conference on Very Large Databases (VLDB'02), Hong Kong, China, 2002, pp. 990-1001.
- [KN10] Kozmina, N., Niedrite, L. 'OLAP Personalization with User-Describing Profiles'. In Proceedings of the 9th International Conference on Perspectives in Business Informatics Research (BIR'10), Rostock, Germany, Springer, Heidelberg, 2010, LNBIP, vol. 64, pp. 188-202.
- [KN11] Kozmina, N., Niedrite, L. 'Research Directions of OLAP Personalization'. In Proceedings of the 19th International Conference on Information Systems

- Development (ISD'10), Prague, Czech Republic. Springer Science+Business Media, 2011, pp. 345-356.
- [KN14] Kozmina, N., Niedrite, L. 'Extending a Metamodel for Formalization of Data Warehouse Requirements'. In: Johansson, B. et al. (eds.) Perspectives in Business Informatics Research, Lund, Sweden, LNBIP, vol. 194, Springer, Berlin, 2014, pp. 362-374.
- [KNG13] Kozmina, N., Niedrite, L., Golubs, M. 'Deriving the Conceptual Model of a Data Warehouse from Information Requirements'. In: Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS'13), Angers, France, vol. 1, 2013, pp. 136-144.
- [KO04] Kaldeich, C., Oliveira, J. 'Data Warehouse Methodology: A Process-driven Approach'. LNCS, vol. 3084, Springer, Heidelberg, 2004, pp. 536-549.
- [KPP02] Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., Rosenberg, J. 'Preliminary Guidelines for Empirical Research in Software Engineering'. IEEE Transactions on Software Engineering, 2002, 28(8): 721-734.
- [KR02] Kimball, R., Ross, M. 'The Data Warehouse Toolkit 2nd Ed: The Complete Guide to Dimensional Modeling'. New York, NY: John Wiley & Sons, Inc., 2002.
- [KR13] Kimball, R., Ross, M. 'The Data Warehouse Toolkit 3rd Ed: The Definitive Guide to Dimensional Modeling'. New York, NY: John Wiley & Sons, Inc., 2013.
- [KS11] Kozmina, N., Solodovnikova, D. '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, 2011, 46:35-42.
- [KS12] Kozmina, N., Solodovnikova, D. 'Towards Introducing User Preferences in OLAP Reporting Tool'. In: Niedrite, L., et al. (eds.) BIR 2011 Workshops, Riga, Latvia. Springer, Heidelberg, 2012, LNBIP, vol. 106, pp. 209-222.
- [KT03] Kelly, D., Teevan, J. 'Implicit Feedback for Inferring User Preference: A Bibliography'. ACM SIGIR Forum, vol. 37, issue 2, 2003, pp. 18-28.
- [LSS05] Lane P., Schupmann V., Stuart I. 'Oracle® Database Data Warehousing Guide 10g Release 2 (10.2)'. Redwood City (CA), Oracle Corporation, 2005, 618 p.
- [LM04] List, B., Machaczek, K. 'Towards a Corporate Performance Measurement System'. In Proceedings of ACM Symposium on Applied Computing (SAC'04), ACM Press, New York, 2004, pp. 1344-1350.
- [Mar12] Marcel, P. 'OLAP Query Personalisation and Recommendation: An Introduction'. In: Aufaure, M.-A. and Zimányi, E. (eds.) eBISS 2011, Springer, Heidelberg, 2012, LNBIP, vol. 96, pp. 63-83.
- [Mar14] Marcel, P. '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.
- [MBI] My Browser Info. [online] <http://mybrowserinfo.com/>
- [MN11] 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)*, Clermont-Ferrand, France, B-7:119-134
- [MSKM07] Mansmann, S., Scholl, M.H., Keim, D.A., Mansmann, F. 'Exploring OLAP Aggregates with Hierarchical Visualization Techniques'. In Proceedings of the 22nd Annual ACM Symposium on Applied Computing (SAC'07), Multimedia & Visualization Track, 2007, Seoul, Korea, pp. 1067-1073.
- [MS08] Mansmann, S., Scholl, M.H. 'Visual OLAP: A New Paradigm for Exploring Multidimensional Aggregates'. In Proceedings of IADIS International Conference on Computer Graphics and Visualization (MCCSIS'08), Amsterdam, The Netherlands, 2008, pp. 59-66.
- [MSST10] Maidel, V., Shoal, P., Shapira, B., Taieb-Maimon, M. 'Ontological Content-based Filtering for Personalised Newspapers: A Method and its Evaluation'. *Online Information Review*, 2010, vol. 34 issue 5, pp. 729-756.
- [MTL] Microsoft Technet Library.  
[online] <http://technet.microsoft.com/en-us/library/cc917644.aspx>
- [NNK11] Niedritis, A., Niedrite, L., Kozmina, N. 'Performance Measurement Framework with Formal Indicator Definitions'. In: Grabis, J. and Kirikova, M. (eds.) *Perspectives in Business Informatics Research*, LNBIP, vol. 90, Springer, Berlin, 2011, pp. 44-58.
- [Par10] Parmenter, D. 'Key Performance Indicators: Developing, Implementing, and Using Winning KPIs'. Jon Wiley & Sons, Inc., 2nd ed., 2010, 320p.
- [Paz99] Pazzani, M.J. 'A Framework for Collaborative, Content-Based and Demographic Filtering'. *Artificial Intelligence Review*, vol. 13, issue 5-6, December, 1999, pp. 393-408.
- [Poe96] Poe, V. 'Building a data warehouse for decision support'. Prentice Hall, 1996, 210p.
- [PAC+07] Pourshahid, A., Amyot, D., Chen, P., Weiss, M., & Forster, A. J. 'Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools'. In Proceedings of the 10th Workshop of Requirement Engineering (WER'07), 2007, pp. 80-91.
- [PCTM03] Poole, J., Chang, D., Tolbert, D., Mellor, D. 'Common Warehouse Metamodel Developers Guide'. Wiley Publishing, 2003, 704p.
- [PFT03] Pu, P., Faltings, B., Torrens, M. 'User-involved preference elicitation'. In *IJCAI'03 Workshop on Configuration*, Acapulco, Mexico, 2003.
- [PS10] Popova, V., Sharpanskykh, A. 'Modeling Organizational Performance Indicators'. *Information Systems*, 2010, 35(4):505-527.

- [PT07] Popova, V., Treur, J. 'A Specification Language for Organizational Performance Indicators'. *Applied Intelligence Journal*, 2007, 27(3):291-301.
- [PT08] Park, Y.-J., Tuzhilin, A. 'The Long Tail of Recommender Systems and How to Leverage It'. Pu, P. et al. (eds.) In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys'08)*, Lausanne, Switzerland, 2008, pp. 11-18.
- [Ric79] Rich, E. 'User Modeling via Stereotypes'. *International Journal of Cognitive Science*, 1979, 3:329-354.
- [Riz09] Rizzi, S. 'Conceptual Modeling Solutions for the Data Warehouse'. In: Erickson, J. (ed.) *Database Technologies: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2009, pp. 86-104.
- [Ros75] Roscoe, J. T. 'Fundamental Research Statistics for the Behavioral Sciences, 2nd ed.' New York: Holt Rinehart & Winston, 1975, 483p.
- [RA10] Romero, O., Abelló, A. 'A Framework for Multidimensional Design of Data Warehouses from Ontologies'. *Data and Knowledge Engineering*, 2010, 69:1138-1157.
- [RALT06] Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J. 'Research in data warehouse modeling and design: dead or alive?'. In *Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP (DOLAP'06)*, ACM Press, New York, 2006, pp. 3-10.
- [RIS94] Resnick, P., Iacovou, N., Sushak, et al. 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews'. In: *ACM 1994 Conference on Computer Supported Cooperative Work*, New York, NY, 1994, pp. 175-186.
- [RKR05] Rashid, A.M., Karypis, G., Riedl, J. 'Influence in Ratings-Based Recommender Systems: An Algorithm-Independent Approach'. In: *Proceedings of the 5th SIAM International Conference on Data Mining*, Newport Beach, CA, USA, 2005, pp. 556-560.
- [RT09] Ravat, F., Teste, O. 'Personalization and OLAP Databases'. Springer US, *Annals of Information Systems*, vol. 3, *New Trends in Data Warehousing and Data Analysis*, 2009, pp. 1-22.
- [Sar99] Sarawagi, S. 'Explaining Differences in Multidimensional Aggregates'. In *Proceedings of the International Conference on Very Large Databases (VLDB'99)*, September 7-10, 1999, Edinburgh, Scotland, UK, pp. 42-53.
- [SG11] Shani, G., and Gunawardana, A. 'Evaluating Recommendation Systems'. *Recommender Systems Handbook*, Ricci, F. et al. (eds.). 2011, pp. 257-294.
- [SG13] Shani, G., and Gunawardana, A. 'Tutorial on Application-oriented Evaluation of Recommendation Systems'. *AI Communications*, IOS Press, The Netherlands, 2013. 26(2):225-236.
- [Sil01] Silverston, L. 'The Data Model Resource Book'. Revised Edition, vol. 1. John Wiley & Sons, USA, 2001, 542p.

- [Sol07] Solodovnikova, D. 'Data Warehouse Evolution Framework'. Proceedings of the 4th Spring Young Researchers Colloquium on Databases and Information Systems, SYRCoDIS'2007, Moscow, Russia, 2007.
- [Sol08a] Solodovnikova, D. 'The Formal Model for Multiversion Data Warehouse Evolution'. In Postconference proceedings of the 8th International Baltic Conference on Databases and Information Systems (DB&IS'08). Frontiers in Artificial Intelligence and Applications by IOS Press, 2008, pp. 91-102.
- [Sol08b] Solodovnikova, D. 'Metadata to Support Data Warehouse Evolution'. In Proceedings of the 17th International Conference on Information Systems Development (ISD'08), Paphos, Cyprus, 2008, pp. 627-635.
- [Sol10] Solodovnikova, D. 'Uz datu noliktavas shēmas evolūciju orientēts vaicājumu definēšanas un attēlošanas rīks', Doctoral thesis, University of Latvia, 2010.
- [SK11] Solodovnikova, D., Kozmina, N. '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, 2011, pp. 363-370.
- [SKKR00] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 'Analysis of Recommendation Algorithms for e-Commerce'. In Proceedings of the 2nd ACM Conference on Electronic Commerce, 2000, pp. 158-167.
- [SKKR01] Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.T. 'Item-based Collaborative Filtering Recommendation Algorithms'. In Proceedings of the 10th International World Wide Web Conference (WWW'10), Hong Kong, 2001, pp. 285-295.
- [SL01] Shearin, S., Lieberman, H. 'Intelligent Profiling by Example'. In: Proceedings of IUI'01, Santa Fe, New Mexico, USA, January 14-17, 2001, pp. 145-151.
- [SM83] Salton, G., McGill, M. 'Introduction to Modern Information Retrieval'. McGraw-Hill Inc., New York, NY, USA, 1983.
- [STL11] Schroder, G., Thiele, M., Lehner, W. 'Setting Goals and Choosing Metrics for Recommender System Evaluation'. In Proceedings of Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces – 2 (UCERSTI 2) at the 5th ACM Conference on Recommender Systems, Chicago, USA, 2011, pp. 78-85.
- [TSM01] Thalhammer, T., Schrefl, M., Mohania, M. 'Active Data Warehouses: Complementing OLAP with Active Rules'. Data & Knowledge Engineering, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 2001, 39(3):241-269.
- [VM03] Vozalis, E., Margaritis, K.G. Analysis of Recommender Systems Algorithms. In Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA'03), Athens, Greece, 2003, pp. 732-745.
- [VM04] Vozalis, M., Margaritis, K.G. 'Enhancing Collaborative Filtering with Demographic Data: The Case of Item-based Filtering'. In Proceedings of the



- 4th International Conference on Intelligent Systems Design and Applications (ISDA'04), Budapest, Hungary, 2004, pp. 361-366.
- [VPF02] Viappiani, P., Pu, P., Faltings, B. 'Acquiring User Preferences for Personal Agents'. Technical Report for American Association for Artificial Intelligence (AAAI Press), 2002.  
[online] <http://liawww.epfl.ch/Publications/Archive/Viappiani2002.pdf>
- [Wei03] Weibelzahl, S. 'Evaluation of Adaptive Systems'. Doctoral thesis, University of Trier, 2003.  
[online] <http://ubt.opus.hbz-nrw.de/volltexte/2004/234/pdf/20030428.pdf>
- [Wes01] Westerman, P. 'Data Warehousing Using the Wal-Mart Model'. Morgan Kaufmann, 2001, 297p.
- [WCC86] Wigton, R.S., Connor, J.L., Centor, R.M. 'Transportability of a Decision Rule for the Diagnosis of Streptococcal Pharyngitis'. Arch Intern Med. 1986, 146(1):81-83.
- [WHH03] Wohlin, C., Höst, M., Henningsson, K. 'Empirical Research Methods in Software Engineering'. In: A.I. Wang, A.I. and Conradi, R. (eds.) Experiences from ESERNET, LNCS, Springer Verlag, 2003, pp. 7-23.
- [WK07] Wrembel, R., Koncilia, C. 'Data Warehouses and OLAP: Concepts, Architecture and Solutions'. IRM Press, 2007, 361 p.
- [WS03] Winter, R., Strauch, B. 2003. 'A Method for Demand-driven Information Requirements Analysis in Data Warehousing Projects'. In Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), Waikoloa, Hawaii, IEEE, USA, 2003, pp. 1359-1365.
- [Zac] The Zachman Framework™ for Enterprise Architecture.  
[online] <http://www.zachmaninternational.com/index.php/the-zachman-framework>
- [Zac03] Zachman, J. A. 'The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing'. Zachman International, 2003.

## APPENDICES

### Appendix 1. Experimentation tasks for student user group.

In terms of these tasks you will have to execute reports in different recommendation modes in order to find required data. Please complete the following tasks using report recommendations (excluding Test task). In each step of the task the data should be found in terms of a single report and written down. **Time interval: 01.02.2012 – 01.05.2012.**

<b>Test task</b> (to be completed <b>without recommendations</b> ). Please provide answers in consecutive order.	
1. Find a Moodle e-course category with the highest number of user sessions in e-learning system. Session length varies from 10 to 60 min and user role is "Guest".	
<b>Course category:</b>	
2. Using this course category find an average hit count in e-learning system. User role is "Guest".	
<b>Average hit count:</b>	
3. Using this course category find a Moodle e-course with the highest number of tests (assessed with a grade) in March, 2012.	
<b>Moodle e-course:</b>	
4. Find the number of part-time students of the faculty that conforms to this Moodle e-course category, who study for a tuition fee and by correspondence.	
<b>Number of students:</b>	

<b>1st task</b> (to be completed in <b>report structure mode</b> ). In terms of this task you have to explore statistical data on students. Please provide answers in consecutive order.	
1. Find a Moodle e-course category where the value of gradebook usage rate is $200 \leq N \leq 300$ .	
<b>Moodle e-course category:</b>	
2. Using this Moodle e-course category find a Moodle e-course with the highest number of interim grades in February, 2012.	
<b>Moodle e-course:</b>	
3. Using this Moodle e-course find an average final grade value and the number of average final grades.	
<b>Average final grade:</b>	<b>Number of grades:</b>
4. In which month of the given period of time the number of tasks in this Moodle e-course was the highest?	
<b>Month:</b>	

<p><b>2nd task</b> (to be completed in <b>semantic mode</b>). In terms of this task you have to explore statistical data on foreign students.</p>	
<p>Before completing the task you have to create user preferences in your profile and set a Degree of Interest (from 0 to 100%) to each term in bold. The more interested you are in it, the higher is the Degree of Interest:</p> <ul style="list-style-type: none"> <li>- Highly interested in <b>foreign students</b> (females by <b>gender</b>);</li> <li>- Interested in <b>level of education, expulsion reason</b>;</li> <li>- Less interested in <b>finance group</b>;</li> <li>- Much less interested in <b>Moodle e-course, Moodle e-course category</b> and <b>average</b> (aggregate function).</li> </ul>	
<p>1. Find a thematic field with the highest number of foreign female students who study full-time.</p>	
<p><b>Thematic field:</b></p>	
<p>2. Find a faculty and the number of expelled Bachelor foreign students in this faculty, where students' expulsion reason was "On one's own initiative".</p>	
<p><b>Faculty:</b></p>	<p><b>Number of students:</b></p>
<p>3. Find Bachelor study programs with foreign students who study for a tuition fee on 01.02.2012.</p>	
<p><b>Study program:</b></p>	
<p>4. Find a Moodle e-course from the Moodle e-course category "Faculty of Chemistry", where foreign students have the highest average final grade.</p>	
<p><b>Moodle e-course:</b></p>	
<p><b>3rd task</b> (to be completed in <b>user activity mode</b>). In terms of this task you have to explore statistical data on full-time students of the Faculty of Computing.</p>	
<p>1. In Moodle e-course category „Computer Science Bachelor” find an average final grade and the number of final grades of foreign students in „DatZ4022: Concepts of Operating Systems” course.</p>	
<p><b>Average final grade:</b></p>	<p><b>Number of grades:</b></p>
<p>2. What is the total number of tasks in this course in March, 2012?</p>	
<p><b>Number of tasks:</b></p>	
<p>3. What is the number of active users in Moodle e-course category „Computer Science Bachelor” whose user role is „Student”?</p>	
<p><b>Number of users:</b></p>	
<p>4. What is the number of students of Doctoral level of education on 01.02.2012 who study for free and whose study program is "Computer Science"?</p>	
<p><b>Number of students:</b></p>	
<p><b>Thank you for taking the time to complete the tasks!</b></p>	

## Appendix 2. Experimentation tasks for academic staff user group.

In terms of these tasks you will have to execute reports in different recommendation modes in order to find required data. Please complete the following tasks using report recommendations (excluding Test task). In each step of the task the data should be found in terms of a single report and written down. **Time interval: 01.02.2012 – 01.05.2012.**

<p><b>Test task</b> (to be completed <b>without recommendations</b>). Please provide answers in consecutive order.</p>
<p>1. Find a Moodle e-course category with the highest number of user sessions in e-learning system. Session length varies from 10 to 60 min and user role is "Guest".</p>
<p><b>Course category:</b></p>
<p>2. Using this course category find an average hit count in e-learning system. User role is "Guest".</p>
<p><b>Average hit count:</b></p>
<p>3. Using this course category find a Moodle e-course with the highest number of tests (assessed with a grade) in March, 2012.</p>
<p><b>Moodle e-course:</b></p>
<p>4. Find the number of part-time students of the faculty that conforms to this Moodle e-course category, who study for a tuition fee and by correspondence.</p>
<p><b>Number of students:</b></p>

<p><b>1st task</b> (to be completed in <b>report structure mode</b>). In terms of this task you have to explore statistical data on students. Please provide answers in consecutive order.</p>
<p>1. Find a faculty where <math>1200 \leq N \leq 1500</math> female students study full-time.</p>
<p><b>Faculty:</b></p>
<p>2. In this faculty find an expulsion reason with the highest number of expelled students.</p>
<p><b>Expulsion reason:</b></p>
<p>3. Using this expulsion reason and this faculty find a level of education.</p>
<p><b>Level of education:</b></p>
<p>4. How many first year full-time students were expelled in this level of education for this expulsion reason?</p>
<p><b>Number of students:</b></p>

<p><b>2nd task</b> (to be completed in <b>semantic mode</b>). In terms of this task you have to explore statistical data on foreign students.</p>	
<p>Before completing the task you have to create user preferences in your profile and set a Degree of Interest (from 0 to 100%) to each term in bold. The more interested you are in it, the higher is the Degree of Interest:</p> <ul style="list-style-type: none"> <li>- Highly interested in <b>foreign students</b> (females by <b>gender</b>);</li> <li>- Interested in <b>level of education, expulsion reason</b>;</li> <li>- Less interested in <b>finance group</b>;</li> <li>- Much less interested in <b>Moodle e-course, Moodle e-course category</b> and <b>average</b> (aggregate function).</li> </ul>	
<p>1. Find a thematic field with the highest number of foreign female students who study full-time.</p>	
<p><b>Thematic field:</b></p>	
<p>2. Find a faculty and the number of expelled foreign students of Bachelor level of education in this faculty, where students' expulsion reason was "On one's own initiative".</p>	
<p><b>Faculty:</b></p>	<p><b>Number of students:</b></p>
<p>3. Find Bachelor study programs with foreign students who study for a tuition fee on 01.02.2012.</p>	
<p><b>Study program:</b></p>	
<p>4. Find a Moodle e-course from the Moodle e-course category "Faculty of Chemistry", where foreign students have the highest average final grade.</p>	
<p><b>Moodle e-course:</b></p>	

<p><b>3rd task</b> (to be completed in <b>user activity mode</b>). In terms of this task you have to explore statistical data on full-time students of the Faculty of Computing.</p>	
<p>1. How many students and foreign students of Bachelor level of education of the Faculty of Computing were expelled?</p>	
<p><b>Number of students:</b></p>	<p><b>Number of foreign students:</b></p>
<p>2. How many students in thematic field „Computer Science” were expelled from the university for the reason „Hasn't completed the study program”?</p>	
<p><b>Number of students:</b></p>	
<p>3. What are the expulsion reasons for students of the Faculty of Computing who study for free?</p>	
<p><b>Expulsion reasons:</b></p>	
<p>4. What is the number of male students of the Faculty of Computing who study for free?</p>	
<p><b>Number of students:</b></p>	
<p><b>Thank you for taking the time to complete the tasks!</b></p>	

### Appendix 3. Experimentation tasks for administrative staff user group.

In terms of these tasks you will have to execute reports in different recommendation modes in order to find required data. Please complete the following tasks using report recommendations (excluding Test task). In each step of the task the data should be found in terms of a single report and written down. **Time interval: 01.02.2012 – 01.05.2012.**

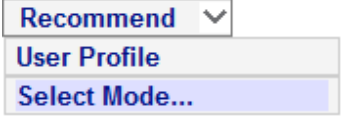
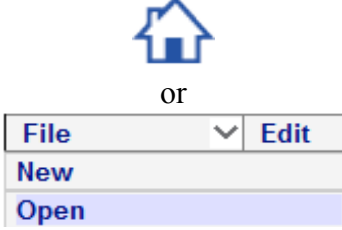
<p><b>Test task</b> (to be completed <b>without recommendations</b>). Please provide answers in consecutive order.</p>
<p>1. Find a Moodle e-course category with the highest number of user sessions in e-learning system. Session length varies from 10 to 60 min and user role is "Guest".</p>
<p><b>Course category:</b></p>
<p>2. Using this course category find an average hit count in e-learning system. User role is "Guest".</p>
<p><b>Average hit count:</b></p>
<p>3. Using this course category find a Moodle e-course with the highest number of tests (assessed with a grade) in March, 2012.</p>
<p><b>Moodle e-course:</b></p>
<p>4. Using this course category find a lecturer that has given the highest number of grades to students in March, 2012.</p>
<p><b>Lecturer:</b></p>

<p><b>1st task</b> (to be completed in <b>report structure mode</b>). In terms of this task you have to explore statistical data on students. Please provide answers in consecutive order.</p>
<p>1. Find a level of education where on 01.05.2012 the number of students who study for free is <math>500 \leq N \leq 525</math>.</p>
<p><b>Level of education:</b></p>
<p>2. How many returning students who study part-time are in this level of education?</p>
<p><b>Number of students:</b></p>
<p>3. Find a faculty where the number of full-time female students who study for a tuition fee is <math>60 \leq N \leq 90</math>.</p>
<p><b>Faculty:</b></p>
<p>4. How many students in this faculty were expelled from the university for the reason „Has not passed final exams“?</p>
<p><b>Number of students:</b></p>

<p><b>2nd task</b> (to be completed in <b>semantic mode</b>). In terms of this task you have to explore statistical data on foreign students.</p>	
<p>Before completing the task you have to create user preferences in your profile and set a Degree of Interest (from 0 to 100%) to each term in bold. The more interested you are in it, the higher is the Degree of Interest:</p> <ul style="list-style-type: none"> <li>- Highly interested in <b>foreign students</b> (females by <b>gender</b>);</li> <li>- Interested in <b>level of education, expulsion reason</b>;</li> <li>- Less interested in <b>finance group</b>;</li> <li>- Much less interested in <b>Moodle e-course, Moodle e-course category</b> and <b>average</b> (aggregate function).</li> </ul>	
<p>1. Find a thematic field with the highest number of foreign female students who study full-time.</p>	
<p><b>Thematic field:</b></p>	
<p>2. Find a faculty and the number of expelled foreign students of Bachelor level of education in this faculty, where students' expulsion reason was "On one's own initiative".</p>	
<p><b>Faculty:</b></p>	<p><b>Number of students:</b></p>
<p>3. Find Bachelor study programs with foreign students who study for a tuition fee on 01.02.2012.</p>	
<p><b>Study program:</b></p>	
<p>4. Find a Moodle e-course from the Moodle e-course category "Faculty of Chemistry", where foreign students have the highest average final grade.</p>	
<p><b>Moodle e-course:</b></p>	

<p><b>3rd task</b> (to be completed in <b>user activity mode</b>). In terms of this task you have to explore statistical data on full-time students of the Faculty of Computing.</p>	
<p>1. What is the number of full-time Bachelor students of the Faculty of Computing who were expelled?</p>	
<p><b>Number of students:</b></p>	
<p>2. How many returning students and foreign returning students of Bachelor level of education are at the Faculty of Computing?</p>	
<p><b>Number of students:</b></p>	<p><b>Number of foreign students:</b></p>
<p>3. What are the expulsion reasons for students of the Faculty of Computing who study for free?</p>	
<p><b>Expulsion reasons:</b></p>	
<p>4. How many foreign female students study in thematic field „Computer Science“?</p>	
<p><b>Number of students:</b></p>	
<p><b>Thank you for taking the time to complete the tasks!</b></p>	

## Appendix 4. User guide for report execution in different recommendation modes.

1	<p>Before completing a <i>Test task</i>, you should turn off recommendations in the reporting tool: <b>Recommend</b> → <b>Select Mode...</b> → <b>Turn on recommendations?</b> → Select <b>No</b> and click <b>Save</b>.</p>										
2	<p>Workbooks contain reports (or worksheets). After saving click <b>Select Report...</b> (or &lt;home&gt; or <b>File</b> → <b>Open</b>) and select a workbook that might contain reports of interest. First report in each workbook is always the default one. A user may switch between reports in a workbook.</p>										
<p>A preview of each report is available. It consists of report headers (rows and columns in crosstabs and columns in tables), page items, and data items (only in crosstabs). A report preview may help a user decide if a certain report seems interesting or not before its execution.</p>											
<p style="text-align: center;"><b>Crosstab Report Objects</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Row</th> <th>Column</th> <th>Data Item</th> <th>Page Item</th> </tr> </thead> <tbody> <tr> <td>Kategorija</td> <td>Pieslēguma vieta Sesijas veids</td> <td>Sesiju skaits</td> <td>Loma</td> </tr> </tbody> </table>			Row	Column	Data Item	Page Item	Kategorija	Pieslēguma vieta Sesijas veids	Sesiju skaits	Loma	
Row	Column	Data Item	Page Item								
Kategorija	Pieslēguma vieta Sesijas veids	Sesiju skaits	Loma								
3	<p>Almost all reports contain time parameters – Date from (<i>No</i>) and Date to (<i>Līdz</i>) – that one has to fill in according to an indicated format (for example, yyyy, dd.mm.yyyy, yyyymmdd, where y is year, m is month, and d is day). Click <b>OK</b> to execute the report.</p>										
<p style="text-align: center;"><b>Select values for the following parameters</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>No:</td> <td><input type="text" value="01.02.2012"/></td> <td>dd.mm.yyyy</td> </tr> <tr> <td>Līdz:</td> <td><input type="text" value="01.05.2012"/></td> <td>dd.mm.yyyy</td> </tr> <tr> <td colspan="3" style="text-align: center;"><input type="button" value="OK"/></td> </tr> </table>			No:	<input type="text" value="01.02.2012"/>	dd.mm.yyyy	Līdz:	<input type="text" value="01.05.2012"/>	dd.mm.yyyy	<input type="button" value="OK"/>		
No:	<input type="text" value="01.02.2012"/>	dd.mm.yyyy									
Līdz:	<input type="text" value="01.05.2012"/>	dd.mm.yyyy									
<input type="button" value="OK"/>											
4	<p>In <i>Task 1</i> select <b>Report Structure</b> mode: <b>Recommend</b> → <b>Select Mode...</b> → <b>Turn on recommendations?</b> → Select <b>Yes</b> → Select <b>Report Structure Mode</b> and click <b>Save</b>.</p>	<p><b>Select Mode...</b></p> <ul style="list-style-type: none"> <li><input type="radio"/> Automatically</li> <li><input checked="" type="radio"/> By Report Structure</li> <li><input type="radio"/> By User Activity</li> <li><input type="radio"/> By Semantic Meaning</li> </ul>									



<p>5</p>	<p>Click <b>Select Report...</b> (or &lt;home&gt; or <b>File → Open</b>) to see a list of workbooks. Select a workbook that might contain reports of interest and repeat step 3.</p> <div style="text-align: center;"> <p><b>Recommendation mode is saved successfully!</b></p> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p>Recommendation mode is <b>On</b>.</p> <p>Your current recommendation mode settings are: <b>Top10</b> recommendations in <b>Structure</b> mode.</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <span style="border: 1px solid gray; padding: 2px 10px;">Back</span> <span style="border: 1px solid gray; padding: 2px 10px; background-color: #e0f0ff;">Select Report...</span> </div> </div> </div>																																												
<p>6</p>	<p>You may control recommendations by clicking <b>Show Recommendation</b> to see the recommendation component or <b>Hide Recommendations</b>. Click the <b>recommendation link</b> (&lt;workbook title&gt;.&lt;report title&gt;) to go to a report of interest. While completing the tasks, try to use recommendation component extensively. However, if recommendations do not help, repeat step 2.</p> <div style="margin-top: 10px;"> <p>Page &lt; 1 / 1 &gt; 1 <span style="margin-left: 20px;">Show</span> <span style="margin-left: 20px; border: 1px solid gray; padding: 2px 5px;">Hide Report Recommendations</span></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Nr.</th> <th style="text-align: left;">Recommendation by User Activity</th> <th style="text-align: left;">Similarity</th> <th style="text-align: left;">Access Date</th> </tr> </thead> <tbody> <tr> <td>1</td> <td><a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa kursu kategorijām</a></td> <td>0.375; 0.338</td> <td>14.07.2014 14:50:27</td> </tr> <tr> <td>2</td> <td><a href="#">Vērtējumu grāmata - Uzdevumu skaits. Kopējais uzdevumu skaits mēnesī pa kursiem</a></td> <td>0.366; 0.181</td> <td>14.07.2014 14:49:20</td> </tr> <tr> <td>3</td> <td><a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem</a></td> <td>0.341; 0.173</td> <td></td> </tr> <tr> <td>4</td> <td><a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem</a></td> <td>0.341; 0.173</td> <td></td> </tr> <tr> <td>5</td> <td><a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa programmām</a></td> <td>0.326; 0.338</td> <td></td> </tr> <tr> <td>6</td> <td><a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem</a></td> <td>0.326; 0.173</td> <td></td> </tr> <tr> <td>7</td> <td><a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem</a></td> <td>0.326; 0.173</td> <td>14.07.2014 14:36:11</td> </tr> <tr> <td>8</td> <td><a href="#">Vērtējumu grāmata - Vērtējumu skaits. Kopējais vērtējumu skaits mēnesī pa kursiem</a></td> <td>0.326; 0.173</td> <td></td> </tr> <tr> <td>9</td> <td><a href="#">Vērtējumu grāmata - Vērtējumu tipi. Vērtējumu tipu sadalījums mēnesī pa kursiem</a></td> <td>0.326; 0.173</td> <td></td> </tr> <tr> <td>10</td> <td><a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Gala vērtējumu vērtības pa kursiem ārzemniekiem</a></td> <td>0.235; 0.357</td> <td>14.07.2014 14:48:49</td> </tr> </tbody> </table> </div>	Nr.	Recommendation by User Activity	Similarity	Access Date	1	<a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa kursu kategorijām</a>	0.375; 0.338	14.07.2014 14:50:27	2	<a href="#">Vērtējumu grāmata - Uzdevumu skaits. Kopējais uzdevumu skaits mēnesī pa kursiem</a>	0.366; 0.181	14.07.2014 14:49:20	3	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.341; 0.173		4	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.341; 0.173		5	<a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa programmām</a>	0.326; 0.338		6	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173		7	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173	14.07.2014 14:36:11	8	<a href="#">Vērtējumu grāmata - Vērtējumu skaits. Kopējais vērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173		9	<a href="#">Vērtējumu grāmata - Vērtējumu tipi. Vērtējumu tipu sadalījums mēnesī pa kursiem</a>	0.326; 0.173		10	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Gala vērtējumu vērtības pa kursiem ārzemniekiem</a>	0.235; 0.357	14.07.2014 14:48:49
Nr.	Recommendation by User Activity	Similarity	Access Date																																										
1	<a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa kursu kategorijām</a>	0.375; 0.338	14.07.2014 14:50:27																																										
2	<a href="#">Vērtējumu grāmata - Uzdevumu skaits. Kopējais uzdevumu skaits mēnesī pa kursiem</a>	0.366; 0.181	14.07.2014 14:49:20																																										
3	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.341; 0.173																																											
4	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem ārzemniekiem</a>	0.341; 0.173																																											
5	<a href="#">Aktivitāte Moodle vidē - Kopējie un vidējie rādītāji. Aktīvo lietotāju vidējā aktivitāte pa programmām</a>	0.326; 0.338																																											
6	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Gala vērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173																																											
7	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumi. Starpvērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173	14.07.2014 14:36:11																																										
8	<a href="#">Vērtējumu grāmata - Vērtējumu skaits. Kopējais vērtējumu skaits mēnesī pa kursiem</a>	0.326; 0.173																																											
9	<a href="#">Vērtējumu grāmata - Vērtējumu tipi. Vērtējumu tipu sadalījums mēnesī pa kursiem</a>	0.326; 0.173																																											
10	<a href="#">Vērtējumu grāmata - Gala un starpvērtējumu vērtības. Gala vērtējumu vērtības pa kursiem ārzemniekiem</a>	0.235; 0.357	14.07.2014 14:48:49																																										
<p>7</p>	<p>In <b>Task 2</b> first fill in a <b>User Profile: Recommend → User profile</b>.</p> <div style="margin-top: 10px;"> <div style="border: 1px solid gray; padding: 5px; width: fit-content;"> <span style="background-color: #e0e0e0; padding: 2px 10px;">Recommend</span> ▾  <span style="background-color: #e0e0ff; padding: 2px 10px;">User Profile</span>  <span style="background-color: #e0e0e0; padding: 2px 10px;">Select Mode...</span> </div> </div>																																												
<p>8</p>	<p>Select a <b>glossary</b> to see the list of concepts and corresponding synonym terms (1 or more). Add the most appropriate term for each concept of interest to your profile by clicking on it and then click &gt;. When all preferred terms are added, click <b>Next &gt;</b>.</p>																																												

	<p><b>Select terms for composition of preferences in user profile</b></p> <p>Choose one of synonym terms that describes best each concept of interest.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Available Items:</b></p> <p>Glossaries: Studiju process</p> <ul style="list-style-type: none"> <li>[-] Studiju process             <ul style="list-style-type: none"> <li>[-] Apakšgrupa                 <ul style="list-style-type: none"> <li>▪ Apakšgrupa</li> </ul> </li> <li>[-] Atskaitīšanas iemesls                 <ul style="list-style-type: none"> <li>▪ Atskaitīšanas iemesls</li> <li>▪ Nemācās (Līm4)</li> </ul> </li> <li>[-] Atskaitīšanas statuss                 <ul style="list-style-type: none"> <li>▪ Atskaitīšanas statuss</li> <li>▪ Nemācās (Līm3)</li> </ul> </li> <li>[-] Ārzemnieks                 <ul style="list-style-type: none"> <li>▪ Ārzemju students</li> <li>▪ Ārzemnieks</li> </ul> </li> </ul> </li> </ul> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>Selected Items:</b></p> <ul style="list-style-type: none"> <li>[-] Agregātfunkcijas             <ul style="list-style-type: none"> <li>[-] AVG                 <ul style="list-style-type: none"> <li>▪ Vidējais</li> </ul> </li> </ul> </li> </ul> </div>																																				
<p>9</p>	<p>Assign an appropriate <b>Degree of Interest</b> (DOI) to each selected term. DOI values vary from 0 (not interested) to 100 (highly interested). Click <b>Finish</b> to save your user profile.</p> <p><b>Select Degree of Interest</b></p> <p>Values of the Degree of Interest (DOI) vary from 0 (not interested) to 100 (highly interested). Terms with DOI=0 will not be saved to user profile.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Glossary</th> <th>Concept</th> <th>Term</th> <th>Degree of Interest</th> </tr> </thead> <tbody> <tr> <td>Agregātfunkcijas</td> <td>AVG</td> <td>Vidējais</td> <td>25 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Dzimums</td> <td>Dzimums</td> <td>100 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Izglītības līmenis</td> <td>Izglītības līmenis</td> <td>75 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Kategorija</td> <td>Moodle kategorija</td> <td>20 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Finansējums</td> <td>Finansējuma grupa</td> <td>0 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Ārzemnieks</td> <td>Ārzemju students</td> <td>95 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Moodle e-kursa nosaukums</td> <td>E-kursa nosaukums</td> <td>20 ▾</td> </tr> <tr> <td>Studiju process</td> <td>Atskaitīšanas iemesls</td> <td>Atskaitīšanas iemesls</td> <td>70 ▾</td> </tr> </tbody> </table> <p style="text-align: center;"> <input type="button" value="Back"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/> </p>	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 ▾
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 ▾																																		
<p>10</p>	<p>Click <b>Select Mode...</b> → Select <b>Semantic Mode</b> and click <b>Save</b>. Repeat steps <b>5-6</b>.</p>																																				
<p>11</p>	<p>In <b>Task 3</b> select <b>User Activity</b> mode: <b>Recommend</b> → <b>Select Mode...</b> → Select <b>User Activity Mode</b> and click <b>Save</b>. Repeat steps <b>5-6</b>.</p>																																				

## Appendix 5. User survey on report execution in different recommendation modes.

Please state your opinion about experimentation task execution using different report recommendation mode in each task. Please tick with  one appropriate box in all questions unless other guidelines are stated.

<b>Questions about the 1st task that you have completed in report structure mode.</b>				
1. How would you evaluate the complexity of the 1st task?				
<input type="checkbox"/> Very easy	<input type="checkbox"/> Easy	<input type="checkbox"/> Average	<input type="checkbox"/> Hard	<input type="checkbox"/> Very hard
2. How would you evaluate the clarity of the 1st task?				
<input type="checkbox"/> Clear	<input type="checkbox"/> Mostly clear	<input type="checkbox"/> Mostly confusing	<input type="checkbox"/> Confusing	
3. In your opinion, did the report recommendations help you complete the 1st task?				
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No	
4. While completing the 1st task, have you used Top3 report recommendation in most of the cases?				
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No	

<b>Questions about the 2nd task that you have completed in semantic mode.</b>				
5. How would you evaluate the complexity of the 2nd task?				
<input type="checkbox"/> Very easy	<input type="checkbox"/> Easy	<input type="checkbox"/> Average	<input type="checkbox"/> Hard	<input type="checkbox"/> Very hard
6. How would you evaluate the clarity of the 2nd task?				
<input type="checkbox"/> Clear	<input type="checkbox"/> Mostly clear	<input type="checkbox"/> Mostly confusing	<input type="checkbox"/> Confusing	
7. In your opinion, did the report recommendations help you complete the 2nd task?				
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No	
8. While completing the 2nd task, have you used Top3 report recommendation in most of the cases?				
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No	

<b>Questions about the 3rd task that you have completed in user activity mode.</b>				
9. How would you evaluate the complexity of the 3rd task?				
<input type="checkbox"/> Very easy	<input type="checkbox"/> Easy	<input type="checkbox"/> Average	<input type="checkbox"/> Hard	<input type="checkbox"/> Very hard
10. How would you evaluate the clarity of the 3rd task?				
<input type="checkbox"/> Clear	<input type="checkbox"/> Mostly clear	<input type="checkbox"/> Mostly confusing	<input type="checkbox"/> Confusing	
11. In your opinion, did the report recommendations help you complete the 3rd task?				
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No	
12. While completing the 3rd task, have you used Top3 report recommendation in most of the cases?				
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No	

<b>General questions.</b>			
13. How would you evaluate your experience with reporting tools in general?			
<input type="checkbox"/> Novice	<input type="checkbox"/> Advanced user	<input type="checkbox"/> Expert	
14. In your opinion, is it easier to complete the tasks employing any of the recommendation modes (1st – 3rd tasks) than to complete the task without any recommendations (Test task)?			
<input type="checkbox"/> Yes	<input type="checkbox"/> Mostly yes	<input type="checkbox"/> Mostly no	<input type="checkbox"/> No
15. While completing which of the tasks have you used the report recommendations most of all? (may tick 1 or 2 answers)			
<input type="checkbox"/> 1st task	<input type="checkbox"/> 2nd task	<input type="checkbox"/> 3rd task	
16. While completing which of the tasks have you received the most precise recommendations? (may tick 1 or 2 answers)			
<input type="checkbox"/> 1st task	<input type="checkbox"/> 2nd task	<input type="checkbox"/> 3rd task	
Comments on your experience with report recommendation modes:			
<b>Thank you for taking the time to complete this survey!</b>			

**Appendix 6. User survey results grouped by user experience.**

Question	Answer	Novice	Advanced user & Expert	Sparklines
1. How would you evaluate the complexity of the 1st task?	Very easy	0,00%	3,33%	
	Easy	16,67%	30,00%	
	Average	23,33%	20,00%	
	Hard	6,67%	0,00%	
	Very hard	0,00%	0,00%	
2. How would you evaluate the clarity of the 1st task?	Clear	26,67%	36,67%	
	Mostly clear	16,67%	16,67%	
	Mostly confusing	3,33%	0,00%	
	Confusing	0,00%	0,00%	
3. In your opinion, did the report recommendations help you to complete the 1st task?	Yes	30,00%	46,67%	
	Mostly yes	16,67%	6,67%	
	Mostly no	0,00%	0,00%	
	No	0,00%	0,00%	
4. While completing the 1st task, have you used Top3 report recommendation in most of the cases?	Yes	6,67%	3,33%	
	Mostly yes	16,67%	26,67%	
	Mostly no	20,00%	16,67%	
	No	3,33%	6,67%	
5. How would you evaluate the complexity of the 2nd task?	Very easy	0,00%	0,00%	
	Easy	6,67%	20,00%	
	Average	26,67%	33,33%	
	Hard	6,67%	0,00%	
	Very hard	6,67%	0,00%	
6. How would you evaluate the clarity of the 2nd task?	Clear	13,33%	23,33%	
	Mostly clear	26,67%	26,67%	
	Mostly confusing	6,67%	3,33%	
	Confusing	0,00%	0,00%	
7. In your opinion, did the report recommendations help you to complete the 2nd task?	Yes	23,33%	26,67%	
	Mostly yes	23,33%	26,67%	
	Mostly no	0,00%	0,00%	
	No	0,00%	0,00%	
8. While completing the 2nd task, have you used Top3 report recommendation in most of the cases?	Yes	6,67%	3,33%	
	Mostly yes	26,67%	33,33%	
	Mostly no	10,00%	16,67%	
	No	3,33%	0,00%	
9. How would you evaluate the complexity of the 3rd task?	Very easy	3,33%	0,00%	
	Easy	23,33%	33,33%	
	Average	13,33%	13,33%	
	Hard	6,67%	6,67%	
	Very hard	0,00%	0,00%	

10. How would you evaluate the clarity of the 3rd task?	Clear	20,00%	40,00%	
	Mostly clear	23,33%	13,33%	
	Mostly confusing	3,33%	0,00%	
	Confusing	0,00%	0,00%	
11. In your opinion, did the report recommendations help you to complete the 3rd task?	Yes	6,67%	6,67%	
	Mostly yes	36,67%	30,00%	
	Mostly no	3,33%	13,33%	
	No	0,00%	3,33%	
12. While completing the 3rd task, have you used Top3 report recommendation in most of the cases?	Yes	0,00%	3,33%	
	Mostly yes	13,33%	13,33%	
	Mostly no	23,33%	20,00%	
	No	10,00%	16,67%	
13. How would you evaluate your experience with reporting tools in general?	Novice	46,67%	0,00%	
	Advanced user	0,00%	40,00%	
	Expert	0,00%	13,33%	
14. In your opinion, is it easier to complete the tasks employing any of the recommendation modes (1st – 3rd tasks) than to complete the task without any recommendations (Test task)?	Yes	20,00%	33,33%	
	Mostly yes	26,67%	20,00%	
	Mostly no	0,00%	0,00%	
	No	0,00%	0,00%	
15. While completing which of the tasks have you used the report recommendations most of all? (may tick 1 or 2 answers)	1st task	10,00%	10,00%	
	2nd task	10,00%	20,00%	
	3rd task	3,33%	3,33%	
	1st & 2nd task	16,67%	16,67%	
	2nd & 3rd task	6,67%	3,33%	
	3rd & 1st task	0,00%	0,00%	
16. While completing which of the tasks have you received the most precise recommendations? (may tick 1 or 2 answers)	1st task	6,67%	16,67%	
	2nd task	23,33%	26,67%	
	3rd task	3,33%	3,33%	
	1st & 2nd task	13,33%	6,67%	
	2nd & 3rd task	0,00%	0,00%	
	3rd & 1st task	0,00%	0,00%	