Towards Next Generation BI Systems: The Analytical Metadata Challenge

Jovan Varga[1], Oscar Romero[1], Torben Bach Pedersen[2], and Christian Thomsen[2]

[1] Universitat Politècnica de Catalunya, BarcelonaTech
Barcelona, Spain ({jvarga|oromero}@essi.upc.edu)
[2] Aalborg Universitet, Aalborg, Denmark ({tbp|chr}@cs.aau.dk)

Abstract. Next generation Business Intelligence (BI) systems require integration of heterogeneous data sources and a strong user-centric orientation. Both needs entail machine-processable metadata to enable automation and allow end users to gain access to relevant data for their decision making processes. Although evidently needed, there is no clear picture about the necessary metadata artifacts, especially considering user support requirements. Therefore, we propose a comprehensive metadata framework to support the user assistance activities and their automation in the context of next generation BI systems. This framework is based on the findings of a survey of current user-centric approaches mainly focusing on query recommendation assistance. Finally, we discuss the benefits of the framework and present the plans for future work.

Keywords: BI 2.0, metadata, user support

1 Introduction

Next generation BI systems (BI 2.0 systems) shift the focus to the user and claim for a strong user-centric orientation. Through automatic user support functionalities, BI 2.0 systems must enable the user to perform data analysis tasks without fully relying on IT professionals designing/maintaining/evolving the system. Ideally, the end user should be as autonomous as possible and the system should replace the designer by providing maximal feedback with minimal efforts. This is even more important if we consider heterogeneous data sources. However, this scenario is yet far from being a reality. A research perspective on this new scenario can be found in [1]. As discussed in the paper, BI 2.0 systems should provide a global unified view of different data sources. To address new requirements, such as dynamic exploration of relevant data sources at the right-time, it is outlined that automatic information exploration and integration is a must. These characteristics raise the need for semantic-aware systems and machine-processable metadata.

Metadata in BI 2.0 systems are needed to support query formulation, relevant source discovery, data integration, data quality, data presentation, user guidance, pattern detection, mappings of business and technical terms, visualization, and any other automatable task that are to be provided by the system. However, current approaches address specific metadata needs in an ad-hoc manner and using customized solutions. A unified global view of the metadata artifacts needed to support the user is yet missing. In this paper we perform a survey to identify what user assistance functionalities should be supported and by means of which metadata artifacts their automation should be enabled. Identifying such metadata is mandatory to enable their systematic gathering, organization, and exploration.

Contributions. We propose a comprehensive metadata framework that supports user assistance activities and their automation in the context of BI 2.0 systems. Specifically, we describe the additional process of the user-system interaction so that the system does not only answer queries but supports the user during the interaction. We identify the main user assistance activities to be supported and the metadata artifacts to be gathered, modeled, and processed to enable the automation of such tasks. These results are based on a survey of specific approaches devoted to provide user assistance on activities like querying and visualization. Finally, we categorize the metadata artifacts to support the user assistance and their processing to enable automation.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents a survey of user-centric approaches. Then, Section 4 defines the metadata framework according to our findings and finally, Section 5 concludes the paper and provides directions for future work.

2 Related Work

The main challenges of user assistance are highlighted in [16]. This paper outlines the increasing need for system support for user analytical tasks in the settings of fast growing, large-scale, shared-data environments. To assist the user with query completion, query correction, and query recommendations, the authors propose meta querying paradigms for advanced query management and discuss the challenge of query representation and modeling. This challenge motivated our research for metadata capturing queries and other related metadata artifacts for user assistance purposes.

Indeed, as discussed in [10], metadata are important for data warehousing users. The article gives a perspective about the metadata benefits for end users' understanding and usage of data warehouse and BI systems, especially for inexperienced ones. The authors present an end user metadata taxonomy consisting of *definitional*, *data quality*, *navigational*, and *lineage* categories. This is a first attempt to characterize end user metadata in BI systems that we aim at extending for a BI 2.0 context.

With the expansion of the Web, analytical data requirements have exceeded traditional data warehouse settings and now entail the incorporation of new and/or external data sources with unstructured or semi-structured data. In this environment, user assistance and its automation are even a greater need but also a challenge. By analyzing the architectural solutions provided in BI 2.0

systems, in the next paragraphs we discuss that metadata are to be one of the key resources for such task.

Some architectural solutions have been recently presented for BI 2.0 (e.g., [1] and [20]). These systems focus on supporting source discovery, data integration, and user guidance for large and often unstructured data sets. However, they do not provide much details about specific metadata artifacts.

[21] proposes the creation of a knowledge base to support data quality-awareness for the user assistance. This knowledge base is to be represented by means of metadata. Furthermore, in the BI Software-as-a-Service deployment model presented in [9], one of the essential business intelligence services is a meta-data service. It defines the business information to support information exchange and sharing among all other services. For metadata handling, these two approaches refer to a specific metadata framework [25].

Finally, the vision of next-generation visual analytics services is presented in [23]. The challenges of visualization and data cleaning, data enrichment and data integration naturally match the goals of next generation BI systems. The authors analyze these challenges for structured data sets but they note that unstructured and semi-structured data represent even greater challenges. As discussed, these tasks raise the need for a common formalism. The metadata are to address such requirements.

Overall, BI 2.0 systems focus on end-to-end architectural solutions and typically pay little attention to describe how they deal with metadata. Indeed, most of these systems just mention the crucial role of metadata for the system overall success. As mentioned, [9] and [21] suggest the usage of Common Warehouse Metamodel (CWM) [25]. Nevertheless, CWM is a standard for interchange of warehouse metadata that provides means for describing data warehouse concepts but the support provided is incomplete for the BI 2.0 metadata artifacts discussed in the following sections (which we refer to as analytical metadata). In this line, the Business Intelligence Markup Language (BIML) Framework [19] presents the automation achieved by using metadata for tasks like data integration, but it does not cover the user assistance perspective. Hence, in order to gain insight on the needed artifacts, in the next section we focus on approaches addressing user assistance tasks, such as query recommendation, and describe in more detail their metadata needs and management.

3 A Survey of User-centric Approaches

As mentioned in the previous section, we subsequently discuss the approaches providing user support functionalities (typically, query recommendation), and primarily focusing on the metadata artifacts used and their exploitation. Thus, we focus on *analytical metadata* meant to support the user analytical tasks.

3.1 Methodology

As our work is motivated by [16], the search for relevant references started with it and the papers citing it / cited by it. We iteratively followed the references found looking for relevant approaches (on journals and conferences) and detecting the most relevant authors in this field. This search was complemented by the keyword searches on the main research related engines. Soon, the area of query recommendation proved to be the most related one and we repeated and refined our searches for relevant papers on this topic. During our search, our primary focus was to detect metadata artifacts used for the user assistance. Therefore, we selected those papers proposing concrete solutions (implementations and/or theoretical foundations for the user assistance) providing enough details about the detection and definition of metadata artifacts. Due to the space limitation, we present a subset of papers representing the whole set of papers found.

3.2 Classification of the Surveyed Approaches

In the typical user-system interaction, the user poses a query, the system processes the query and returns the query result to the user. Throughout this process the user often needs assistance. In the reviewed approaches, we encountered various forms of user support. Figure 1 describes the additional process flows triggered to provide such support and refers to the main *metadata artifacts* collected and exploited.

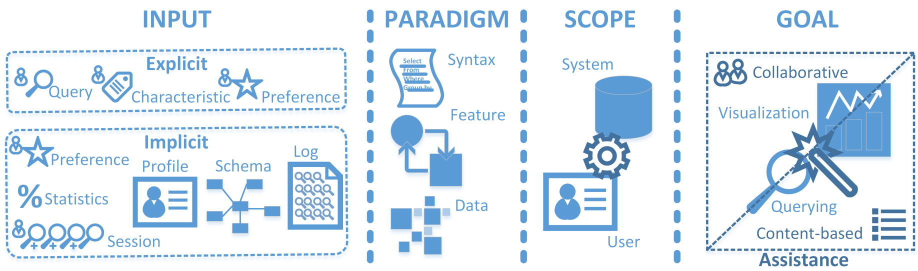

Fig. 1. Assistance Process

The system first gathers the needed **input** to be processed in order to achieve the ultimate **goal** of the user support (e.g., query recommendation). We further classify this process according to the level of abstraction or **paradigm** used to process the input data (syntactic, feature-based or data-based approaches) and the process **scope** (profiling the user, the system or both). Table 1 shows how the reviewed approaches were classified.

The **input** includes *metadata artifacts* that are defined in Table 2. As illustrated in Figure 1, we distinguish between *explicit* and *implicit* input. *Explicit* input represents data currently produced by the user that triggers the user assistance process of the system, whereas *implicit* input refers to input that was previously gathered (and stored in the metadata repository), either coming from the system or the user, as well as further metadata inferred from both. *Explicit*

Table 1. Classification of the reviewed approaches

Approach	Explicit Input	Implicit Input	Paradigm	Scope	Goal	Assistance Techniques
SQL QueRIE Recommendations [4]	Query	Query Log, User Session	Syntax, Data	User Profiling	Querying	Collaborative
Similarity Measures for OLAP Sessions [5]	Query	User Session, Schema	Syntax, Feature (Schema)	/	Querying	Content-based
Predicting Your Next OLAP Query Based on Recent Analytical Sessions [6]	Query	Query Log, Schema	Syntax, Feature (Schema)	System Profiling	Querying	Content-based
A Personalization Framework for OLAP Queries [7]	Query, Preferences	User Profile	Feature (Preferences)	User Profiling	Visualization	Content-based
Query Recommendations for Interactive Database Exploration [8]	Query	Query Log, User Session	Data	User Profiling	Querying	Collaborative
Expressing OLAP Preferences [12]	Query, Preferences	Schema	Feature (Preferences)	User Profiling	Querying	Content-based
myOLAP: An Approach to Express and Evaluate OLAP Preferences [13]	Query, Preferences	Schema	Feature (Preferences)	User Profiling	Querying	Content-based
SnipSuggest: Context-Aware Autocompletion for SQL [17]	Query	Query Log	Syntax	System Profiling	Querying	Content-based
The Meta-Morphing Model Used in TARGIT BI Suite [22]	Query	Statistics, Preferences, Schema	Feature (Preferences, Statistics)	User Profiling	Querying, Visualization	Content-based
"You May Also Like" Results in Relational Databases [30]	Query	Query Log, Schema	Data, Feature (Schema)	User Profiling, System Profiling	Querying	Content-based, Collaborative
Recommending Join Queries via Query Log Analysis [31]	Query	Query Log	Syntax	System Profiling	Querying	Content-based
A Framework for Recommending OLAP Queries [11]	Query	Query Log, User Session	Syntax	System Profiling	Querying	Content-based
Meet Charles, Big Data Query Advisor [28]	Query	Statistics	Feature (SDL, Statistics)	System Profiling	Querying	Content-based

input detected in our survey are queries, preferences, and user characteristics. *Implicit* input are previously logged queries, typically stored in the sequence they were posed (i.e., sessions), stored user profiles, automatically inferred user preferences, detected system statistics, and the system schema.

Queries are the main input considered in all surveyed approaches. Processing queries is primarily performed at three abstraction levels or **paradigms**: at the *syntax* level, at the *data* level, or modeled according to a certain *feature*. According to [16], *syntactic* processing happens when the syntactic structure of the query is the main facet to be explored (e.g., to combine fragments and propose new queries). For example, [6] presents a framework for recommending OLAP queries based on a probabilistic model of the user's behavior, which is computed by means of query similarities at the syntactic level. *Data-based* processing describes the query in terms of the data it retrieves. For example, in [8] the user queries are characterized by the retrieved tuples. Alternatively, *feature-based* processing models and stores the query in terms of a certain *feature. Features* encountered in the reviewed approaches are schema information, user preferences (and visual constraints), statistics, and ad-hoc languages to capture semantic fragments from the queries (e.g., the Segmentation Descrip-

Table 2. Analytical Metadata Artifacts

Metadata Artifact	Definition	Example
Query	The user inquiry for certain data, disregarding the form it takes	What is the total quantity per product and location?
Preferences	The result set selection and/or representation prioritization	Preferred results of sales where amount is in between 1000 and 5000 range; Preferred representation is a pie chart
User characteristics	Explicitly stated data characterizing the user	Job position, department, office location
Query log	The list of all queries ever posed	{Query 1, query 2, query 3, query 4, ..., query N}
User session	Automatically detected sequence of queries posed by the user when analyzing or searching for certain data	<Query 1, query 3, query 7>
User profile	The set of user characteristics and preferences	Characteristics: User id '1', job position 'manager' Preferences: Preferred monthly over quarter overview
Statistics	Automatically detected data usage indicators	Product id 'P' searched in 23% (12345) of cases
System schema	The data model of the system	Dimension Tables: ProductDimension, DateDimension, LocationDimension Fact Tables: SalesFact

tion Language in [28]). Interesting conclusions can be drawn for each paradigm. *Syntactic* approaches lack semantics and suffer from several drawbacks. First, differently formulated queries returning the same data cannot be identified as equivalent. Second, the natural interconnection between a series of queries in a single analytical session is lost (or cannot be easily represented). Third, the data granularity produced cannot be detected (as in general, a pure syntactic approach is performed). All in all, exploitation is limited due to the usage of recorded syntactic artifacts only. *Data-based* approaches characterize queries according to the data they retrieve, which entails similar deficiencies due to the lack of semantics gathered. Similar queries returning disjoint results due to some filtering conditions or data aggregation cannot be identified. Also, their interconnection cannot be easily represented. Nevertheless, the main deficiency of this paradigm is its questionable feasibility in the context of BI 2.0 systems, which typically consider Big Data settings with large amounts of data.

None of the two previous paradigms are powerful and flexible enough to fully capture the intention of the user and detect similar queries in a broader sense. This is the main goal behind the last paradigm, which opens new possibilities for addressing this challenge. The concept of a *feature* focuses on modeling the input query to gain additional semantics representing meaningful information that previous paradigms miss. Several current approaches can be classified according to these terms. For example, [5], [6], and [30] represent the queries in terms of the schema. Moreover, [22] proposes the use of the multidimensional model to both model the system schema and the queries posed. For recommendations, it uses recorded user actions and preferences or predefined settings. However, the recommendation potential based on multidimensional semantics such as hierarchical dimension organization is not fully exploited. Finally, other approaches such as [12, 13] introduce the means for the user to explicitly express her preferences when querying the data. However, the user is assumed to manually express the preferences.

The process **scope** defines the profiling need for user assistance purposes. We consider two scope types, *user profiling* that correlates (input) metadata artifacts with the user and *system profiling* that creates a general set of (input) metadata artifacts about the system. While metadata generated in both cases are then used for multiple user assistance purposes, most approaches focus on *user profiling* and few pay attention to *system profiling*, which opens new interesting possibilities, such as self-tuning systems.

The ultimate **goal** and the final step in the assistance process is the concrete user assistance produced. There are multiple forms of user assistance related to the different phases of user-system interaction. We generalize them into two major categories. The first category is *querying* assistance which covers various forms of user support when querying a database. The most typical querying assistance is query recommendation, but other tasks such as query completion, result selection, result recommendations, and join recommendations are identified in our survey. The second major category of user assistance is *visualization*. This assistance refers how to represent the query output in the most satisfactory way for the user. Although undoubtedly crucial for BI 2.0, little attention has been paid to this issue.

An orthogonal aspect to the two categories just discussed are the **assistance techniques** used to provide support. We typically talk about *collaborative* techniques, which entail assistance generation based on the metadata gathered for similar users, and *content-based* techniques, which only exploit the metadata related to a certain user for providing support.

Finally, since many ideas for database recommendation systems come from web solutions we aim at completing our survey by briefly positioning web recommender systems (e.g., see [2]) in terms of our classification. For generating personalized recommendations, web recommendation approaches typically rely on *user profiles*. They process user *queries* at the *data* level, i.e., according to the results retrieved, and profile users and, to some extent, systems. On this base, they provide *content-based*, *collaborative*, or hybrid recommendations. Additionally, as suggested in [2] and thoroughly elaborated in [3], current web recommendation systems should take into account context information for generating context-aware recommendations. In terms of our classification, the context is generally covered by either *user characteristics* and *user preferences*, or by data itself (e.g., in BI, the time/location information are typically covered by appropriate dimensions). Relevantly, web-based recommender systems strongly rely on *user characteristics* (mostly ignored in the database field) and profiles.

As result, by analyzing the described *user assistance process* (Figure 1), we identified currently used *metadata artifacts* (Table 2), outlined the importance of a *feature-based* approach for gathering and modeling metadata artifacts, remarked that *system profiling* can be used for more than just user assistance and highlighted the importance of *user characteristics* and profiling metadata.

4 The Analytical Metadata Challenge

The survey presented currently used metadata artifacts, their handling, and exploitation potential. Nevertheless, in the context of BI 2.0 systems there is a need for the automation of user assistance activities. Therefore, in this section, we propose a comprehensive framework to address this challenge. First, we present the metadata artifacts to be gathered in the Analytical Metadata (AM) repository and then we discuss how to gather, model, and process them in order to automate their management.

4.1 Analytical Metadata

AM are the set of metadata artifacts entailed by BI 2.0 systems to support the user decision making process. To clarify these artifacts we extend the end user metadata taxonomy from [10] as illustrated in Figure 2. The original taxonomy includes *definitional*, *data quality*, *navigational*, and *lineage* metadata categories that are business oriented and do not refer to technical artifacts. We define the technical interpretation of these categories, add a new category, and classify concrete metadata artifacts accordingly. The *definitional* category defines the integration schema, user characteristics, and a vocabulary of business terminology. The *data quality* category describes data set characteristics. The *navigational* category keeps evidence about how the user explores and navigates data. The *lineage* category captures the origin of data including data sources, transformations, and mappings. The *ratings* category covers metadata artifacts about user interests and data usage statistics.

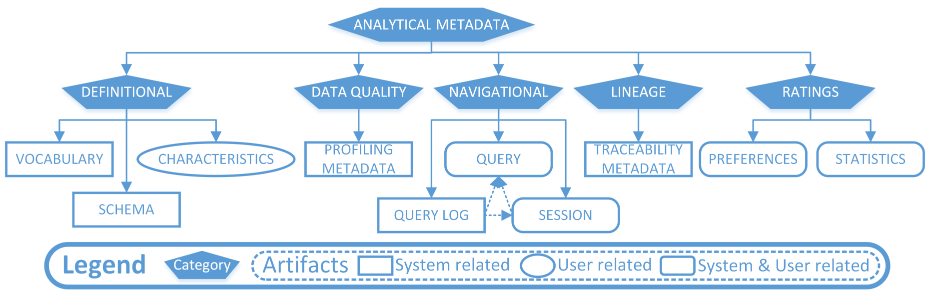

Fig. 2. Analytical Metadata Taxonomy

The schema, characteristics, query, session, query log, preferences, and statistics artifacts were introduced in Section 3. In this section, we classify these artifacts and introduce new ones to enhance automation. The metadata artifacts may refer to the system (e.g., schema), the user (e.g., characteristics) or both (e.g., a query defines user interests and refers to the schema structure).

Definitional. The *vocabulary* and *schema* artifacts are fundamental for all the other categories, i.e., all other metadata artifacts should be defined in terms of them. *Vocabulary* defines business terms, their relationships, and their mappings to the integration schema. Its primary role is to act as a reference terminology where to map all gathered metadata artifacts. It can efficiently be represented with an ontology [29] that is machine-processable and enables the automatic reasoning needed for the automation of user support. Next, in the context of BI systems, we propose the *schema* to be represented by means of the multidimensional (MD) model [18]. As discussed in [26], the MD model is mature and well-founded and has key applicability in data warehousing, on-line analytical processing (OLAP), and increasingly in data mining. It captures analytical perspectives by means of facts and dimensions. In this context, it defines necessary constraints and is covered by the MD algebra [27] that together determine potential user actions. For example, if a user analyzes data on a `Month` level, just based on the *schema* she can be suggested to change the granularity to the `Day` or `Year` level even if no one performed this analysis before. Lastly, the *user characteristics* artifact is borrowed from web recommender systems and defines the user by capturing explicitly asserted information that cannot be automatically detected (e.g., job position, age, etc.). It is typically stored as unstructured data and if defined in terms of the *vocabulary* it can be used for metadata processing, e.g., as parameter for recommending algorithms, pattern detection, etc.

Data quality. To tackle *data quality* from a technical point of view, we propose metadata profiling processes (e.g., as explained in [24]) to gain insight into data. *Profiling metadata* characterize data sets like value range, number of values, number of unique values, sparsity, and similar metrics. This way data can be automatically annotated so that domain experts are provided with quality evidences for the data used. For example, inaccurate analytical results might come from data sparsity, i.e., a non-representative data sample.

Navigational. In compliance with the MD representation suggested for the *schema*, we propose the MD model as modeling feature (see previous section for further details) used to capture *queries*, *logs*, and *sessions*. Moreover, to better capture the user intentions, we propose to represent *queries* as ETL flows. As elaborated in [15], a query can be represented as a directed graph of operators. In turn, each operator is characterized as its input and output schema, which should follow the MD principles. This solution is more generic than a typical query definition and enables representation of more complex transformations (e.g., *rollup*), supports lineage, and can be represented as a graph that facilitates manipulations in comparison to declarative queries. Although more powerful, managing such a complex representation is more demanding (e.g., computing similarities or containment between ETL flows) and remains as an open challenge.

Lineage. For *lineage*, we propose the *traceability metadata* artifact that must capture the information about data sources, transformations performed when migrating data from the sources, and mappings to the integration *schema* (e.g., see [14,15]). This way the system may provide the user with explanations about how an analytical value is computed and from what sources.

Ratings. The *user preferences* artifact can be manually stated, e.g., using a preference algebra [13]. Although this option might suit advanced users, whenever possible, it is preferable to automatically detect them by appropriate processing techniques over other metadata artifacts. For example, we may infer from the *queries* gathered that the user systematically applies some filtering predicates when navigating the data. Finally, the *statistics* artifact represents data usage indicators that can be considered as a kind of query profiling, i.e., to keep evidence about what data are more explored, as well as more complex indicators (e.g., the most popular combinations of fact and dimension tables [22]).

4.2 Automation and Processing

For the efficient management and storage of the AM we alternatively categorize its artifacts into the categories illustrated in Figure 3. These categories elaborate on how to gather each artifact, its level of processing automation, and exploitation purposes. This categorization must be used to guide the AM storing organization.

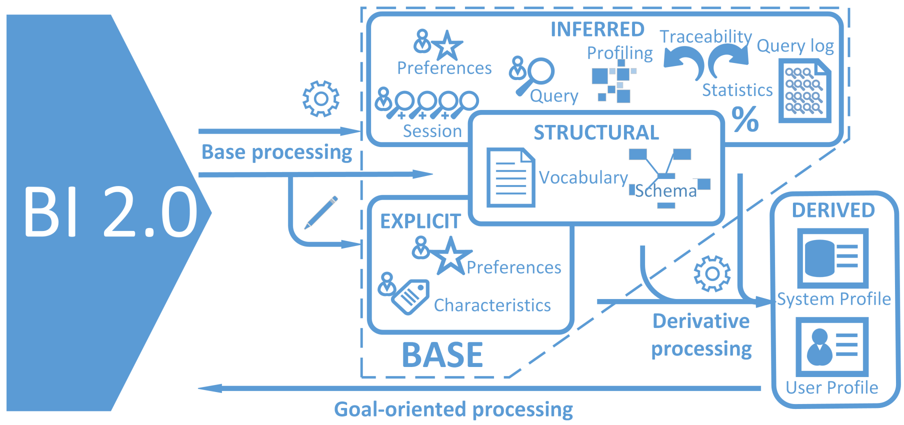

Fig. 3. Analytical Metadata Processing Categories

In the categorization, we talk about **structural**, **explicit**, **inferred**, and **derived** metadata. As discussed earlier, the *schema* and *vocabulary* artifacts are fundamental for all the other artifacts and for this reason we refer to them as structural artifacts. Typically, the system designers define and maintain these artifacts. Explicit artifacts (i.e., *user preferences* and *characteristics*) are those explicitly stated by the user and not automatically detected by the system. Contrary, inferred artifacts are automatically gathered by the system and thus, they can be automatically detected and stored without the explicit help of the user. This category mainly refers to *query logs* and *sessions*, but also to automatically detected *preferences*, data usage *statistics*, as well as the *profiling* and *traceability* metadata artifacts. Inferred artifacts are feature-based, i.e., either modeled according to the MD *schema* or in terms of the definitional *vocabulary*. The structural, explicit, and inferred categories jointly represent the minimal set of information about user actions and interests to be gathered for automating user assistance within the system. For this reason, we refer to these three categories as the **base** metadata, which determine the exploitation possibilities of the AM.

The remaining **derived** metadata category results from processing base metadata according to certain exploitation purposes (typically user and/or system profiling). The produced *user/system profiles* are materialized pieces of derived metadata typically aimed at improving the performance of the algorithms used to provide user assistance. For example, a collaborative recommending system requires to compute similar users. Computing similar users must be performed from base metadata artifacts and it can be previously materialized as derived metadata (i.e., user profiling) in order to improve the response time of the recommending system. Note that system profiling opens new possibilities for system self-tuning capabilities.

Automation implies processing flows (denoted as arrows in Figure 3) to populate and exploit the AM artifacts. Our goal here is not to define concrete algorithms but to point out the metadata management and processing flows to be considered when implementing the metadata repository. Consequently, we talk about **base**, **derivative**, and **goal-oriented processing**. The base processing populates the base metadata (i.e., the structural, explicit, and inferred artifacts) by interacting with the BI system. Specifically, the explicit metadata is stated by the user, whereas inferred artifacts are automatically detected and gathered from the system. Structural metadata is typically maintained by the system administrator. Contrary, the derivative processing populates the derived metadata from the base metadata gathered (i.e., derivative processing happens within the AM repository). Finally, goal-oriented processing exploits both the base and derived metadata in order to give concrete user assistance. It represents the purpose of the AM storage and exploitation. The aims of goal-oriented processing are, for example, query recommendations or visualization techniques, whereas final products are recommended queries, graphs/charts, and potentially self-tuning database actions for database optimizations (e.g., data usage *statistics* can serve to trigger indexing of some frequently used attribute).

All in all, due to the expected large volumes of metadata in these systems and the demanding processing capabilities described, the AM repository should be implemented in a dedicated subsystem.

5 Conclusions and Future Work

We have presented a comprehensive metadata framework to support user assistance and its automation in the context of BI 2.0 systems. The framework is based on the findings of a survey of existing user-centric approaches where we describe the user assistance process and identify assistance activities and metadata artifacts needed. It proposes an AM repository by categorizing the metadata artifacts to support the user assistance and their processing to enable automation. By introducing the subsets of automatically inferred and derived metadata artifacts with corresponding processing techniques, our framework motivates and directs the automation of the user assistance process which is one of the key requirements of BI 2.0 systems. As metadata artifacts are described on a high abstraction level, the framework is a base to support user assistance features over BI 2.0 heterogeneous data sources. Moreover, since AM also capture the information about the system usage, they can serve for other purposes typically overlooked, such as system self-tuning and optimization.

In our future work, we plan to define the metamodel of AM and provide an implementation of the AM support for query recommendation.

Acknowledgments. This research has been funded by the European Commission through the Erasmus Mundus Joint Doctorate "Information Technologies for Business Intelligence - Doctoral College" (IT4BI-DC) and it has been partly supported by the Spanish Ministerio de Ciencia e Innovación under project TIN2011-24747.

References

1. Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J.N., Naumann, F., Pedersen, T.B., Rizzi, S., Trujillo, J., Vassiliadis, P., Vossen, G.: Fusion Cubes: Towards Self-Service Business Intelligence. IJDWM 9(2), 66–88 (2013)
2. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE TKDE 17(6), 734–749 (2005)
3. Adomavicius, G., Tuzhilin, A.: Context-Aware Recommender Systems. In: Recommender Systems Handbook, pp. 217–253. Springer (2011)
4. Akbarnejad, J., Chatzopoulou, G., Eirinaki, M., Koshy, S., Mittal, S., On, D., Polyzotis, N., Varman, J.S.V.: SQL QueRIE Recommendations. PVLDB 3(2), 1597–1600 (2010)
5. Aligon, J., Golfarelli, M., Marcel, P., Rizzi, S., Turricchia, E.: Similarity Measures for OLAP Sessions. KAIS pp. 1–27 (2013)
6. Aufaure, M.A., Kuchmann-Beauger, N., Marcel, P., Rizzi, S., Vanrompay, Y.: Predicting Your Next OLAP Query Based on Recent Analytical Sessions. In: DaWaK. pp. 134–145 (2013)
7. Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D.: A Personalization Framework for OLAP Queries. In: DOLAP. pp. 9–18 (2005)
8. Chatzopoulou, G., Eirinaki, M., Polyzotis, N.: Query Recommendations for Interactive Database Exploration. In: SSDBM. pp. 3–18 (2009)
9. Essaidi, M.: ODBIS: Towards a Platform for On-Demand Business Intelligence Services. In: EDBT/ICDT Workshops (2010)
10. Foshay, N., Mukherjee, A., Taylor, A.: Does Data Warehouse End-User Metadata Add Value? Commun. ACM 50(11), 70–77 (2007)
11. Giacometti, A., Marcel, P., Negre, E.: A Framework for Recommending OLAP Queries. In: DOLAP. pp. 73–80 (2008)

12. Golfarelli, M., Rizzi, S.: Expressing OLAP Preferences. In: SSDBM. pp. 83–91 (2009)
13. Golfarelli, M., Rizzi, S., Biondi, P.: myOLAP: An Approach to Express and Evaluate OLAP Preferences. IEEE TKDE 23(7), 1050–1064 (2011)
14. Jovanovic, P., Romero, O., Simitsis, A., Abelló, A., Mayorova, D.: A Requirement-Driven Approach to the Design and Evolution of Data Warehouses. IS In Press (2014)
15. Jovanovic, P., Romero, O., Simitsis, A., Abelló, A.: Integrating ETL Processes from Information Requirements. In: DaWaK. pp. 65–80 (2012)
16. Khoussainova, N., Balazinska, M., Gatterbauer, W., Kwon, Y., Suciu, D.: A Case for A Collaborative Query Management System. In: CIDR (2009)
17. Khoussainova, N., Kwon, Y., Balazinska, M., Suciu, D.: SnipSuggest: Context-Aware Autocompletion for SQL. PVLDB 4(1), 22–33 (2010)
18. Kimball, R., Reeves, L., Ross, M., Thornthwaite, W.: The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses. John Wiley & Sons, Inc. (1998)
19. Leonard, A., Masson, M., Mitchell, T., Moss, J., Ufford, M.: Business Intelligence Markup Language. In: SQL Server 2012 Integration Services Design Patterns, pp. 301–326. Springer (2012)
20. Löser, A., Hueske, F., Markl, V.: Situational Business Intelligence. In: Business Intelligence for the Real-Time Enterprise, pp. 1–11. Springer (2009)
21. Mazón, J.N., Zubcoff, J.J., Garrigós, I., Espinosa, R., Rodríguez, R.: Open Business Intelligence: On the Importance of Data Quality Awareness in User-Friendly Data Mining. In: EDBT/ICDT Workshops. pp. 144–147 (2012)
22. Middelfart, M., Pedersen, T.B.: The Meta-Morphing Model Used in TARGIT BI Suite. In: ER Workshops. pp. 364–370 (2011)
23. Morton, K., Balazinska, M., Grossman, D., Mackinlay, J.D.: Support the Data Enthusiast: Challenges for Next-Generation Data-Analysis Systems. PVLDB 7(6), 453–456 (2014)
24. Naumann, F.: Data Profiling Revisited . SIGMOD Record 32(4) (2013)
25. Object Management Group: Common Warehouse Metamodel Specification 1.1 (last accessed January, 2014), http://www.omg.org/spec/CWM/1.1/PDF/
26. Pedersen, T.B.: Multidimensional Modeling. In: Encyclopedia of Database Systems, pp. 1777–1784. Springer US (2009)
27. Romero, O., Abelló, A.: On the Need of a Reference Algebra for OLAP. In: DaWaK. pp. 99–110 (2007)
28. Sellam, T., Kersten, M.L.: Meet Charles, Big Data Query Advisor. In: CIDR (2013)
29. Skoutas, D., Simitsis, A.: Ontology-Based Conceptual Design of ETL Processes for Both Structured and Semi-Structured Data. IJSWIS 3(4), 1–24 (2007)
30. Stefanidis, K., Drosou, M., Pitoura, E.: "You May Also Like" Results in Relational Databases. In: PersDB Workshop (2009)
31. Yang, X., Procopiuc, C.M., Srivastava, D.: Recommending Join Queries via Query Log Analysis. In: ICDE. pp. 964–975 (2009)