An Analytical Model for Data Persistence in Business Data Warehouses

Veit Köppen [#1], Thorsten Winsemann [*2], Gunter Saake [#3]

[#] *Institute for Technical and Business Information Systems, Otto-von-Guericke University Magdeburg*
Universitätsplatz 2, 39106 Magdeburg, Germany
[1] veit.koeppen@ovgu.de
[3] gunter.saake@ovgu.de

[*] *SAP*
Germany
[2] thorsten.winsemann@t-online.de

***Abstract*—Redundancy of data persistence in Data Warehouses is mostly justified with better performance when accessing data for analysis. However, there are other reasons to store data redundantly, which are often not recognized when designing data warehouses. Especially in Business Data Warehouses, data management via multiple persistence levels is necessary to condition the huge amount of data into an adequate format for its final usage. Redundant data allocates additional disk space and requires time-consuming processing and huge effort for complex maintenance. That means in reverse: avoiding data persistence leads to less effort. The question arises: What data for what purposes do really need to be stored? In this paper, we discuss decision support and evaluation approaches beyond cost-based comparisons. We use a compendium of purposes for data persistence. We define a model that includes objective indicators and subjective user preferences for decision making on data persistence in Business Data Warehouses.**

We develop an indicator system that enables the measurement of technical as well as business-related facts. With multi-criteria decision methodology, we present a framework to objectively compare different alternatives for data persistence. Finally, we apply our developed method to a real world Business Data Warehouse and show applicability and integration of our model in an existing system.

I. Introduction

Data Warehouses (DW), especially Business Data Warehouses (BDW), are often characterized by enormous data volumes [1]. When designing and operating a BDW, there are high requirements regarding data provision, such as performance, granularity, flexibility, and timeliness.

Besides, restrictions call for additional, redundant data. For example, materialized views and summarization levels are commonly used for enhancing speed of data access.

Performance is still the main reason for storing data redundantly, but there are several other reasons which lead to additional data persistence, as for instance design decisions, usability, or security. Such reasons are often underestimated or even not considered, but frequently motivating data storage. Yet, additional storage always requires a huge effort to guarantee consistency and limits prompt data availability. Latest technology, based on in-memory databases (IMDB), improves response times for data access and implies to enable non-redundant data access without any loss of performance [2]. Assuming that performance of data access - as the main reason for storing data redundantly - is less important in IMDB, the question arises, which data persistence is still necessary in BDW? As less persistent data means less effort, the necessity of persistence has to be questioned. In order to identify such necessities, we present a compendium of purposes for data persistence and use it as a basis for decision-making whether to store data or not. Moreover, demands on BDW are changing and evolving continuously. So, the need for existing persistent data has to be questioned occasionally. With this background, we discuss decision support approaches to define which data are needed persistently. Our approach goes beyond cost-based comparisons and also includes user preferences by using methods of multi-criteria decision analysis [3]. Note, such considerations are valid for BDW based on relational databases, too; however, the lower performance of the database motivates additional data storage anyway.

In this paper, we extend our work in [4] and present a formal methodology for evaluating persistence in BDW. With the help of a formalization of measuring and evaluation technical and business-driven indicators as well as user preferences in a holistic comparison, we decide for data persistence in an objective way. Our findings are presented with an artificial example that shows the general applicability of our approach. Furthermore, we evaluate our approach with a real-world case study, which is adapted to the systems specifics and therefore, this evaluation additionally shows applicability to for another scenario.

This paper is organized as follows: In Section II, we briefly introduce BDW specialties and the layered architecture. This section is based on our findings in [5] and are used as background for the domain of our decision model. In Section III, we present a compendium of reasons for data persistence in BDW and we define a process for decision support of data persistence. Both sections build the basis for our formal extension from [4] and [5] that we present in this paper. In Section IV, we define a ratio system as a basis for decision-making, enhance it with means of multi-criteria decision analysis (MCDA) methods and present an approach for an evaluation model for data persistence. In Section V, we show and discuss evaluation results of our persistence model with the help of a real-world case study. Section VI gives an overview on related work. Finally, Section VII concludes our paper with an outlook on future work.

II. Business Data Warehouses

In this section, we briefly introduce characteristics of a BDW. Additionally, we present a layered architecture for BDW and classify the layers to the DW reference architecture.

A. Characteristics of a BDW

A BDW [6] is a DW to support decisions concerning the business on all organizational levels. It covers all business areas, such as logistics, finance, and controlling. Moreover, it is an important basis for applications, such as business intelligence, planning, and customer relationship management. A BDW collects and distributes huge amounts of data from a multitude of heterogeneous source systems. As it provides a single version of truth for all companys data, there must be a common view on centralized, accurate, harmonized, consistent, and integrated data at a given point in time. The range of use is often world-wide. So, data from different time-zones have to be integrated. Frequently, 24x7-hours data availability has to be guaranteed, facing the problem of loading and querying at the same time. In addition, there are high requirements on BDW data: ad-hoc access, near real-time availability, high data quality, and the need for very detailed and granular data with a long time horizon. Moreover, new or changing requirements of BDW users have to be flexibly and promptly satisfied.

B. A Layered Architecture for BDW

Persistence in DW is closely connected to the architecture. That means, the decision to store data comes along with the datas format and the area or layer, where data are stored. Excluding data sources, the common reference architectures (e.g., [7], [8]) define three main areas, representing three aspects of data handling: data acquisition in the staging area, data processing in the basis database, and data provision in the data marts. Within this rather rough model, persistence on each level is implicit [9]. Another classification for data warehouse architectures is given in [10]. A decision on the architecture is presented by the authors from an organizational view. We restrict ourselves to the question of data persistence, which is applicable to all architectures in different ways.

Regarding BDW, a layered architecture, as introduced by [11], refines the three areas approach (see. Fig. 1). Herein, layers become more detailed and dedicated. Each of the five layers represents an area for increasing the datas value with respect to the usage. That means, for instance, that a data set does not have to be lifted to the highest level of data marts (and stored there), if it is already usable (e.g., for reporting) on a lower level. Yet, a layer does not imply data storage by definition. One has to consider the data format and where to store data. Though, persistence has to be decided prior based on purposes for data needs. We briefly describe the individual areas of a layered architecture in the following.

The *Data Acquisition Layer* represents the extraction phase, the "inbox" of the warehouse, where incoming data are accepted usually without modification. In the *Quality & Harmonization Layer*, data are integrated technically and semantically, including de-duplication, aspects of information integration and so on; that is, transformation within "conventional" ETL process. At the *Data Propagation Layer*, the companys data are kept as a single version of truth of harmonized and integrated data, without any business logic; therefore, it defines a common data basis for all applications. In the *Business Transformation Layer*, data are transformed due to business needs, which can be dependent on different department requirements; for instance, order and invoice data are combined for computing open orders information. At the *Reporting & Analysis Layer*, data are transformed mainly according to requirements of usage (e.g., computing rolling periods values) and to enable fast access to the data. Within the Operational Data Provider, data are simply transformed for specific business cases (e.g., near real-time reporting).

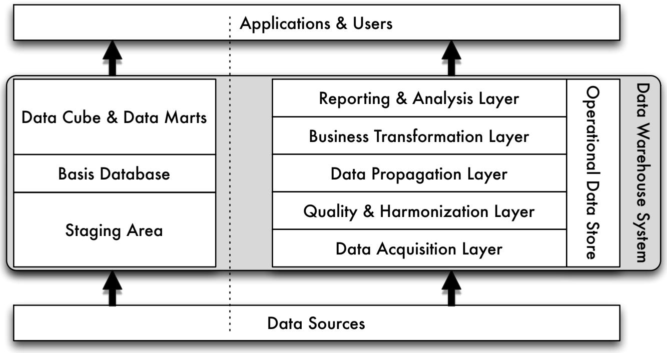

Fig. 1. Common DW-Architecture and Layered Architecture for BDW

Although the boundaries are shifting, a rough classification of these layers to the three warehouses aspects of data handling can be done: Data acquisition is covered within the Data Acquisition and the Quality & Harmonization Layer, data processing in the Data Propagation and Business Transformation Layer, and data provision in the Reporting & Analysis Layer. We give a more detailed description of a layered architecture and its comparison to reference architectures in [5].

III. Reasons for Data Persistence

Besides operating a BDW in a given architectural layout, we define reasons for data persistence in this section. Furthermore, we present a classification schema and depict a step-by-step model for BDW persistence.

Mainly there are two reasons for persistence in DW: storage of transformed data in the basis database and storage of redundant, aggregated data in the data mart layer. However, there is a broad range of further reasons for storing data in a DW system [12]. They can be based on technical conditions, companies terms of governance, legal restrictions. Moreover, the ease of data maintenance is another purpose as well as simply subjective needs for safety or security.

A. Data Persistence in BDW

In the following, we present reasons for data persistence; see [12] for a detailed description with examples. For structuring purposes, we arrange these reasons into the following five areas, namely data acquisition, data modification, data management, data availability, and laws and provisions.

1) *Data acquisition*: source system decoupling, data availability, extensive data recreation, data lineage
2) *Data modification*: changing transformation rules, addicted transformations, complex data transformation, complex different data representation
3) *Data management*: constant data basis, en-bloc data supply, complex authorization, single version of truth (SVoT), corporate data memory (CDM)
4) *Data availability*: information warranty, data access performance
5) *Laws and provisions*: corporate governance, laws and provisions

The operation of productive DW necessarily means potential for conflicts which arises from requirements of data usage as well as time and effort to create necessary prerequisites. Persistence often means redundant data, because source and transformed target data sets are stored at the same time. As transformations are usually not unique, keeping only the target data means a loss of information. Such redundancy leads to huge additional effort. Firstly, it concerns hardware aspects such as additionally allocated disk space. Even more important, it impacts the period of time that is required for data processing in the BDW, which includes creation and maintenance of consistent data sets. Another important aspect is the additional amount of work that is necessary by the responsible DW and database administrators.

B. Mandatory vs. Essential Persistence

A decision for persisting data cannot just be made cost-based, for instance by comparing disk space and updating cost versus gain in performance. One has to take the purpose of the data storage into account, to define whether it is helpful, rather essential, or even mandatory. In order to identify the necessity of persistence, we classify such reasons into two groups: mandatory and essential persistence. Mandatory persistence applies to data that have to be stored according to laws and regulations of corporate governance. It also holds for data that cannot be replaced because they are not available any longer or cannot be reproduced due to changes of transformation rules. Lastly, data that are required for other data transformations must be stored if simultaneous availability is not ensured. Essential persistence can be additionally classified into certain categories. Firstly, data that are available or reproducible in principle, however, the effort for reproducing is quite high. Another group is data which are stored to simplify the maintenance or operation of the DW or related applications. A third group of data is persistent due to the DW conceptual design: SVoT and CDM; cf. [12]. Fourthly, responsibility for guaranteeing information leads to data storage for safety reasons. Finally, data are redundantly stored for performance purposes – often the largest volume. For a complete grouping of persistence purposes by necessities, we refer to [12]. Table I. In this table, we also name the corresponding groups as categories for essential persistence. Column "Area" shows the relevant area mentioned above. Column "Step" as sort key represents the process step number we present in Fig. 2 and describe in the following.

TABLE I. PERSISTENCE PURPOSES, GROUPED BY NECESSITIES

Area	Purpose	Necessity	Category	Step
2	Addicted transformation	Mandatory	-	1
5	Corporate governance	Mandatory	-	1
5	Laws and provision	Mandatory	-	1
1	Data availability	Mandatory	-	2
2	Changing transformation rules	Mandatory	-	3
3	Single version of truth	Essential	Design	4
3	Corporate data memory	Essential	Design	4
1	Source system decoupling	Essential	Recreation	5
1	Extensive data recreation	Essential	Recreation	5
1	Data lineage	Essential	Simplification	6
2	Complex, different data	Essential	Simplification	6
3	Constant data basis	Essential	Simplification	6
3	En-bloc data supply	Essential	Simplification	6
3	Complex authorization	Essential	Simplification	6
4	Information warranty	Essential	Safety	6
2	Complex data transformation	Essential	Performance	7
4	Data access performance	Essential	Performance	7

C. Decision for Data Persistence

Although the question which data to store is valid for BDW based on any database, it becomes urgent with more powerful systems. The approach of column-oriented databases (cf. [13], [14]), came into special focus in the DW community (e.g., [15], [16]), because of the advantages regarding data compression and read access [17]. Today, there are commercially offered IMDB, which are used for DW applications (e.g., [18], [19]). These changes within technology lead to the question, in which degree persistence in IMDB-based DW is required or necessary. [20] and [21] suggests to store no data additional to the source data and to compute any data on-the-fly. Here, all data come into focus, which are not stored mandatory. Especially, this concerns data that have been additionally stored for enhancing access performance or due to complex transformations. That does not mean that all additional persistence is obsolete due to processing speed in such systems – as to en-bloc data supply or the creation of a constant data basis for planning, it will be even more valid.

A decision to store data must take into consideration the reason for persistence and its necessity. For instance, a regulation drives into persistence; basically, this applies for all mandatory stored data. The decision is more difficult for essentially stored data, as the reason cannot be quantified clearly. This is valid for categories recreation, simplification, safety, and performance. Solely design reasons are identifiers for data storage. Fig. 2 shows a decision flow for persistence of any data. It is simplified, because rather fuzzy terms, such as "complex" and "frequently", have to be specified dependent on the domain and application scenario. The first three steps deal with mandatory stored data. For essentially stored data except "design", decision indicators are much diversified.

Persistent data in BDW systems require effort for maintenance among others. In order to avoid dispensable data persistence, the need for such persistence has to be defined by the purpose of the data. Therefore, we classify reasons for persistence in such systems. Based on this, we come up to decide whether to store data or not. For mandatory stored data, this decision is clear. However, the need for essentially stored data is more difficult to decide and goes beyond pure cost-based comparisons of system data. We present an approach that combines system data with user workload. Preferences of decision-makers are also considered by including methods of MCDA to be able to support decision-making. Our approach

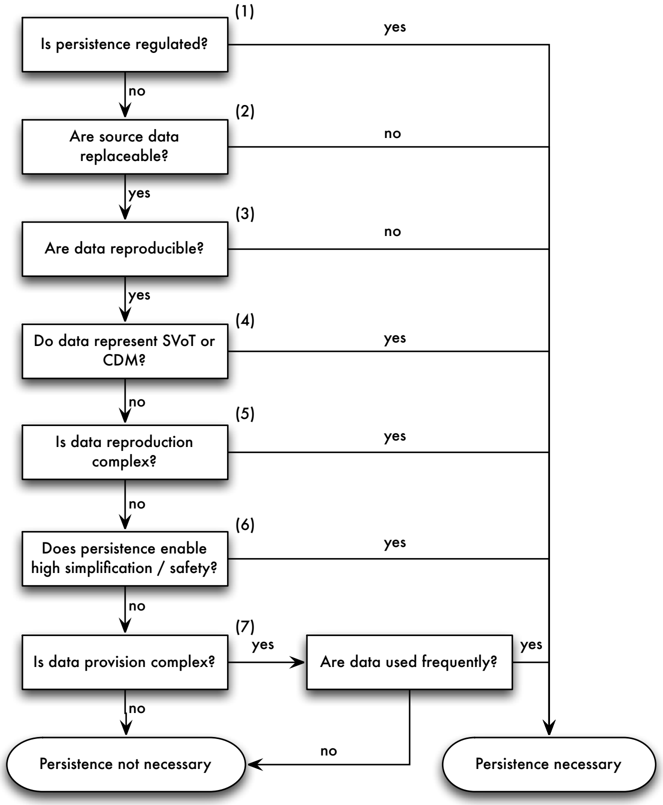
(1)
Is persistence regulated?
yes
no
(2)
Are source data replaceable?
no
yes
(3)
Are data reproducible?
no
yes
(4)
Do data represent SVoT or CDM?
yes
no
(5)
Is data reproduction complex?
yes
no
(6)
Does persistence enable high simplification / safety?
yes
no
(7)
Is data provision complex?
yes
Are data used frequently?
yes
no
no
Persistence not necessary
Persistence necessary

Fig. 2. Decision Process for Persistence

particularly addresses IMDB in BDW systems.

In this paper, we extend and formalize our decision model to evaluate different data cubes, data marts, or other persistence, including the definition of a proper set of formulas. We also include report variants that cannot be operated by a cube due to missing elements (cf. Section IV). Moreover, we perform extensive tests, including real-life data. Although we focus on formalization within steps 5-7 in our model (see Fig. 2) in more detail in Section IV, we briefly discuss the decision steps 1-4 in the following.

The first three steps deal with mandatory data, which are not under consideration as they must be stored. For essentially data, the answer whether to store is much diversified. When the BDW design includes a corporate data memory, these data have to be stored. In case the reproduction or provision of data is complex (in time and/or resources), factors such as data volume, frequency of data access and data changes, or the speed of data provision have to be taken into account. Besides, the basis for decisions change and data persistence has often more than one reason. For example, as the connection to one source system is excellent, data would not be stored solely due to source system decoupling. Yet, as the transformation is partly complex, data are stored anyhow.

IV. Decision Making for Data Persistence

As already mentioned, factors used as decision indicators for essentially stored data are diverse. Storing data redundantly in so-called aggregates or materialized views is not new and by now quite common for enhancing the speed of data access (e.g., [22], [23], [24], [25], [26]). Therefore, we pick the category "Performance" to discuss this topic in more detail. In this context, we define performance as the speed of data provision. Fig. 3 shows a simple data flow that we use to illustrate the factors used in our ratio system. As such factors are determinable and measurable we denote them as key indicators. Data cube C is filled with new data on a regular basis. This cube is data source for an aggregate cube A (data flow CA); all elements (i.e., dimensions and measures) of A are elements of C, and C is a proper subset of $A : C \subset A$. Report R can therefore be built from data cube C (data flow CR) or from aggregate cube A (data flow AR).

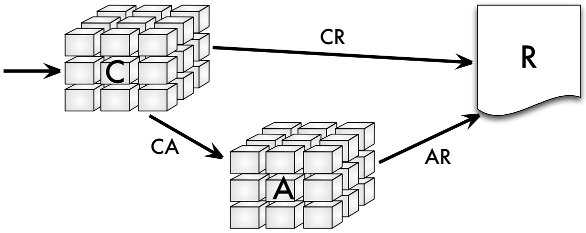

Fig. 3. Data Flow Example

A. Ratio System for Decision Making

The basic question for category "Performance" is: Shall data be stored redundantly to enable higher access performance or shall they be re-created from detailed data when needed? In Fig. 2, it is shown as "Is data provision complex?". Here, "complex" goes beyond a cost-based comparison of disk space usage or updating costs versus gain in performance. In order to quantify "complex", We define an estimation model that consists of four areas, namely data supply, data actualization, data reorganization, and cost. Each of these areas has to be measured in a certain period of time and is described by a set of key indicators as we explain in the following.

We use a classification of our indicators into four classes:

- Data supply,
- Data actualization,
- Data reorganization, and
- Cost.

We identify these classes from our experience as the most prominent ones. However, our model is not restricted to these classes and can be adapted accordingly. Fig. 4 represents our indicator system in a hierarchical description and we describe this in more detail in the following.

Data supply contains time duration T_S and frequency F_S of making data available, for instance when calling a report. T_S is further split into time of querying $T_{Q(S)}$ (i.e. to select the data on the database), the time of transformation $T_{\tau(S)}$ (i.e. to process the data on the application server), and OLAP time $T_{O(S)}$ (i.e. to format the data for reporting). All times and therefore T_S are measured in time units, e.g., minutes or seconds. Setting i as the number of calls in a certain period, we define the total time of data supply for this period as:

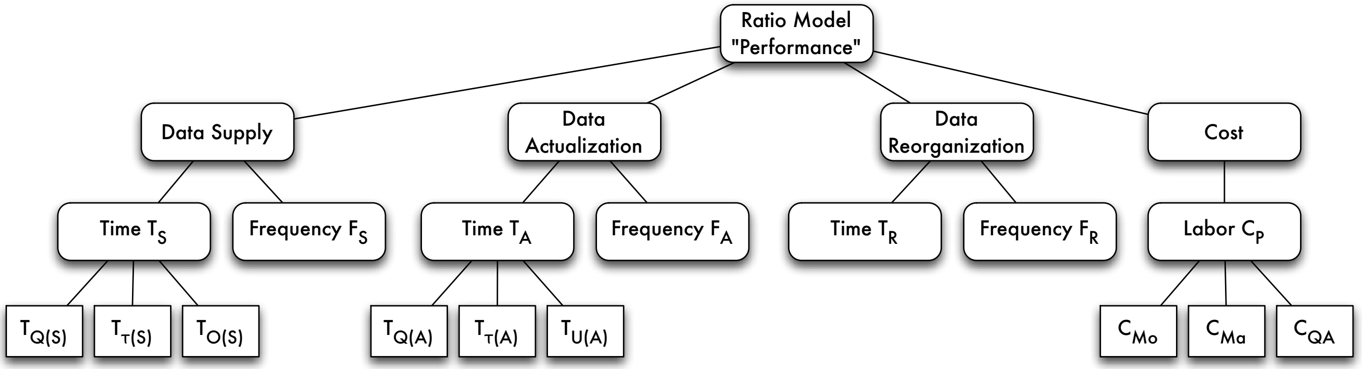

Fig. 4. Ratio System Hierarchy

$$\begin{aligned} Data\ Supply &= \sum_{i=1}^{Calls} F_{S_i} \cdot T_{S_i} \\ &= \sum_{i=1}^{Calls} F_{S_i} \cdot \left(T_{Q(S_i)} + T_{\tau(S_i)} + T_{O(S_i)}\right) \end{aligned} \quad (1)$$

Data actualization contains time duration T_A and frequency F_A of supplying a set of data with new data. T_A is further split into time of querying $T_{Q(A)}$ (i.e. to select the new data on the database), the time of transformation $T_{\tau(A)}$ (i.e. to process the new data on the application server) and updating time $T_{U(A)}$ (i.e. to actualize the existing data set). All times and therefore T_A are measured again in time units. Setting j as the number of loads with new data in a certain period, we define the total time of data supply as:

$$\begin{aligned} Data\ Actualization &= \sum_{j=1}^{Loads} F_{A_j} \cdot T_{A_j} \\ &= \sum_{j=1}^{Loads} F_{A_j} \cdot \left(T_{Q(A_j)} + T_{\tau(A_j)} + T_{U(A_j)}\right) \end{aligned} \quad (2)$$

Data reorganization contains time duration T_R and frequency F_R of reorganizing data. For simplification, we do not divide T_R into the single durations of each data set. Note, for instance index selection in a DW is a time consuming task including labor effort and creation of new index instances with respect to data loads, cf. [27], [28]. T_R is measured in time units. Setting k as the number of reorganization runs in a certain period, we define the total time of data reorganization as:

$$Data\ Reorganization = \sum_{k=1}^{Runs} F_{R_k} \cdot T_{R_k} \quad (3)$$

Cost are mainly labor cost C_P and accumulate for all manpower necessary to operate a DW (e.g., DW modeler or database administrator). These cost can be divided for modeling work C_{Mo} (e.g., re-modeling a data cube), for maintenance work C_{Ma} (e.g., control loading jobs and re-start failed ones), and for quality assurance work C_{QA} (e.g., check data consistency). Moreover, one can also take technical cost into account, which means cost for hardware and electricity. All costs are measured in terms of money. For simplification, we focus on labor cost and define the total cost in a certain period as:

$$Cost = C_P = C_{Mo} + C_{Ma} + C_{QA} \quad (4)$$

The first three areas contain key figures that can be determined directly from the DW system; each of them can be measured in time units. Cost are more complex to determine. For instance, one has to take the required working time for the particular person and the respective salary. As in our model, technical cost are not included, we can simplify by measuring cost in time units, too. Fig. 4 displays our hierarchical system. An adequate period of time for validation is "month" as it involves some advantages. Peaks that can occur on days or weeks are avoided, and cost often are accounted monthly, and quarter or year is too long-term.

Whereas the measurements and derived indicators can be used separately for reasoning for data persistence, we depict in the following section how to normalize all indicators, which is a prerequisite for a fair comparison.

B. An Estimation System for Decision Making

The results of the formulas above are interpreted as "the higher, the worse" – of course, a shorter run time is preferable to a longer one. Yet, we want to make the results of our estimation model more comprehensible and comparable. That means, the results must be interpretable as "the higher, the better". Additionally, the results should be weighted for guaranteeing a sound comparison. Therefore, we define a weighted ratio W. Given a number of x data cubes, the weighted ratio W regarding time (T) for any cube Γ is:

$$W_{T(\Gamma)} = \frac{(\sum_{i=1}^{x} T_i) - T_\Gamma}{\sum_{i=1}^{x} T_i} \quad (5)$$

Regarding cost (C), the weighted ratio is accordingly:

$$W_{C(\Gamma)} = \frac{(\sum_{i=1}^{x} C_i) - C_\Gamma}{\sum_{i=1}^{x} C_i} \quad (6)$$

TABLE II. RUN TIMES AND FREQUENCIES OF REPORT CALLS

RV	$T_{S(A)}(s)$	$F_S(d)$	$T_{S(B)}(s)$
R_1	2.5	20	0.5
R_2	3	5	0.5
R_3	4	10	1
R_4	5	1	1.5
~~R_5~~	~~7.5~~	~~3~~	~~∞~~
$\sum T \cdot S$	110		24

Furthermore, we require that the sum of the weights for all cubes is equal to 1. This enables a user-friendly interpretation of the different alternatives outcomes.

C. An Example with Figures: Using Measures

For ease of understanding, we present a simplified example with arbitrary defined indicators in the following. In practice, these values are measured either directly in the database system or with the help of data warehouse monitors. For more details, we refer to Section V for SAP Hana as one example.

With regard to data supply, we define: Report R can be called with several options (e.g., parameters) and therefore, includes variants (RV): R_1, R_2, R_3, R_4, and R_5. Per day, those variants are called in a certain frequency F_S; dependent of being built from cube A or B, the run time duration (in seconds) is either $T_{S(A)}$ or $T_{S(B)}$. Here, we do not split this time as detailed as in Equation 1, but simply use time duration T_S accordingly.

Table II shows exemplary figures. Note that value "∞" in cell "$R_5/T_{S(B)}$" means that this variant cannot be operated by cube B due to missing elements. Such products are not taken into account for further computations.

According to Equation 5, the weighted ratios for the cubes are: $W_{S(A)} = 0.1791$ (i.e. $(134 - 110)/134$) and $W_{S(B)} = 0.8209 = (134 - 24)/134$.

We ignore time and frequency for data actualization for cube A (i.e. $T_{A(A)} = F_{A(A)} = 0$) as it is required for cube B, too. The actualization time for cube B, we set to $T_{A(B)} = 40s$ with a frequency of $F_{A(B)} = 12$ per day. Here too, we do not split actualization time as detailed as in Equation 2 but simply use time duration T_A. According to Equation 5, the weighted ratios for the cubes are: $W_{A(A)} = 1 = (480 - 0)/480$ and $W_{A(B)} = 0 = (480 - 480)/480$. Again: Note that $\sum W_A = W_{A(A)} + W_{A(B)} = 1$.

Data reorganization occurs once a day for both cubes (i.e. $F_R = 1$); the particular times are: $T_{R(A)} = 10s$ and $T_{R(B)} = 100s$.

According to Equation 5, the weighted ratios for the cubes are: $W_{R(A)} = 0.90909 = (110 - 10)/110$ and $W_{R(B)} = 0.09091 = (110 - 100)/110$. Again, note that $\sum W_R = 1$.

TABLE III. COST RATIOS

	C_{Ma}	C_{Mo}	C_{QA}
Cube A	1	1	1
Cube B	4	3	5
$\sum$	5	4	6
$W_{(A)}$	0.8	0.75	0.83333
$W_{(B)}$	0.2	0.25	0.16667
#Cost	1	1	3

Regarding costs, we do not use absolute values, but define ratios between costs for cubes C and A. Table III shows such cost ratios for maintenance C_{Ma}, modeling C_{Mo}, and quality assurance C_{QA}, listed for cubes C and A in Rows 1 and 2. Row 3 contains the respective sums, and Rows 4 and 5 the weighted ratio according to Equation 6.

As we are dealing with different cost ratios (#Cost = 3), an additional step is necessary to define a weighted sum W^*_{Cost} for all cost ratios:

$$W^*_{Cost} = \frac{\sum W_{Cost}}{\#Cost} \tag{7}$$

The sum of the single weighted ratios for cubes A and B according to Equation 7 are: $W^*_{Cost(A)} = 0.79449 = 2.38333/3$ and $W^*_{Cost(B)} = 0.20556 = 0.61667/3$. Again: Note that $W^*_{Cost} = 1$ for all cubes.

So, we are able to do an evaluation E of the cubes using a simple average of all single results as follows:

$$E = \frac{\sum_x W_x}{\#Criteria} \tag{8}$$

This leads to the following comparison of cube A and B:

$$E(A) = 0.72066 = (0.1791 + 1 + 0.90909 + 0.79449)/4$$
$$E(B) = 0.27934 = (0.8209 + 0 + 0.09091 + 0.20556)/4$$

These results are interpretable in such a manner, that cube A is preferred to cube B; that means cube B is obsolete. However, as the result is equally weighted, it does not reflect possible preferences of users, administrators, or management. For instance, a management directive can be to deliver fast reports prior to any other parameter. Yet, such preferences can be considered using methods of multi-criteria decision analysis. We present this methodology in the following.

D. Multi-Criteria Decision Analysis

In this section, we briefly describe Multi-criteria decision analysis (MCDA) as a tool to evaluate different scenarios as well as different user preferences. We select MCDA due to the fact that:

- it can be easily understood,
- it is easily applicable, and
- it can be reproduced.

MCDA, a sub-discipline of operations research, considers multiple criteria in decision-making processes. MCDA "models do not try to compute an optimal solution, but they try to determine via various ranking procedures either a ranking of the relevant actions (...) that is optimal with respect to several criteria, or they try to find the optimal actions amongst the existing solutions (...) That is, given a set of alternatives and a set of decision criteria, then what is the best alternative?" [29]. MCDA methods are used in various applications, see for example [30], [31].

There is a plurality of different methods of MCDA, of which the cost-utility analysis [32] and the "Analytic Hierarchy Process" (AHP) [33] are commonly used. Both methods measure intangibles, which directly addresses our domain. We do not discuss differences, pros and cons of both methods in detail. However, as the utility analysis impresses by its ease of use, the AHP is more precise by enabling to evaluate all criteria against each other. In our example, we use a mixture of both methods that combines ease of use with the possibility to compare the criteria among each other.

E. An Example with Figures: including User Preferences

In our first example (see Fig. 3), we use a scale of nine values for the paired comparison, see also [34]. The values listed below represent the adjective that replaces "..." within the following comparison: "Criterion 1 has ... importance compared to Criterion 2":

- 1 = equal
- 3 = moderate
- 5 = strong
- 7 = very strong or demonstrated
- 9 = extreme

Interim values express nuances, reciprocals represent "inverted" preferences.

TABLE IV. Criteria Validation/Comparison

	DS	**DA**	**DR**	**Cost**	$\sum c$	WF_c
DS	1	6	7	5	19.000	0.44225
DA	1 / 6	1	7	1 / 6	8.333	0.19397
DR	1 / 7	1 / 7	1	1 / 7	1.429	0.03325
Cost	1 / 5	6	7	1	14.200	0.33053
$\sum\sum c$					42.962	1

Table IV shows the compared pairs of four criteria, namely data supply (DS), data actualization (DA), data reorganization (DR), and cost. That means, for instance: "Data supply has very strong importance compared to data reorganization" (cell "DS/DR = 7"); cell "DR/DS" represents the inverted preference with 1/7. Moreover, column "WF_c" (weighted factor of each criterion) is calculated by formula:

$$WF_c = \frac{\sum_i c_{ij}}{\sum_j \sum_i c_{ij}} \tag{9}$$

Finally, the weighted ratio for each area x (i.e. data supply, data actualization, data reorganization, and cost; see Section IV), has to be rated considering the expressed preferences per criterion WF_c to get an extended evaluation E_x for each cube:

$$E_x = \sum_{i=1}^{x} W_i \cdot WF_c \tag{10}$$

This leads to: $E_{x(C)} = 0.56599$ and $E_{x(A)} = 0.43401$. Note, these evaluations refer to each area, in our model to the four areas data supply, data actualization, data reorganization, and cost. Due to the fact that we do not restrict the model and evaluation method to these four areas, this method can be adapted or extended to further influences or decision criteria.

The figures show that although proportions have changed compared to the ones of our simple example (0.72066 → 0.56599 and 0.27934 → 0.43401), the result remains the same: Cube A is obsolete. However, it can change as soon as preferences change. For instance, the speed of data supply for reporting is defined as extremely important. That means: "data supply has extreme importance compared to all other criteria" (values in cells "DS/DA", "DS/DR", and "DS/Cost" = 9; inverse values respectively 1/9). As a result, ratios for cubes C and A change to 0.49769 and 0.50231, so that cube A is necessary to fulfill the preference of fast reporting.

Within this section, we present a formal decision model using MCDA including measurements directly accessible from a DW system, business-related indicators for including management level decisions, and user preferences, to address specific requirements on an objective basis. In the next section, we show how we achieve applicability of our model in a real-world example.

V. Evaluation

In this section, we evaluate our decision model including user preferences with a real world example. Due to space limitations, we focus on an excerpt of the DW. For more details see also [35]. Firstly, we present our example and show afterward a decision for persistence of materialized cubes. In this section, we show the practical feasibility of our proposed method and at the same time we show that our decision model is easily integrated in a daily used system.

A. Description of the Example

Within this case study, we demonstrate the applicability of our decision model and show which details are required for a sound evaluation.

We use an example from a real world application in the domain of sales and distribution. We only describe an excerpt of the DW. However, the addressed issues are completely covered within our example. Note, for simplicity we restrict the decision domain to three alternatives, which is without loss of generality. In our case study, we also focus on processing steps that occur repetitive and finally, we decide which data persistence is applied on the defined structures.

In our example, we use factual data as decision basis. Note, this restriction is not necessary for our model and it can also be used for raw data, which means an inclusion of all ETL processes, or aggregated cubes or in other words materialized view data. In practice, a restriction of reports that are important and critical is often applied to master the decision on persistence. For this purpose, a grouping of similar reports can be done. This enables at the same time a more balanced system, due to a more robust measurement basis. We also use such a selection in our case study. Finally, we restrict the decision makers, who are involved in the process to two to three persons. This is quite common in practice, because only key-user and DW administrator are involved in the decision on persistence [35].

We perform our case study in a DW that is implemented in SAP NetWeaver Business Warehouse (SAP BW). We measure the technical evaluation numbers as real data directly from the system. That means, we use actual determined run time measurements instead of acceptable numbers by the DW users. Note, for these measurements exist no standardized tests or evaluation benchmarks such as [36]. Therefore, our data represent measurements from a daily in use data warehouse. So, our data are only applicable within this case study and we cannot judge on the corresponding data warehouse system.

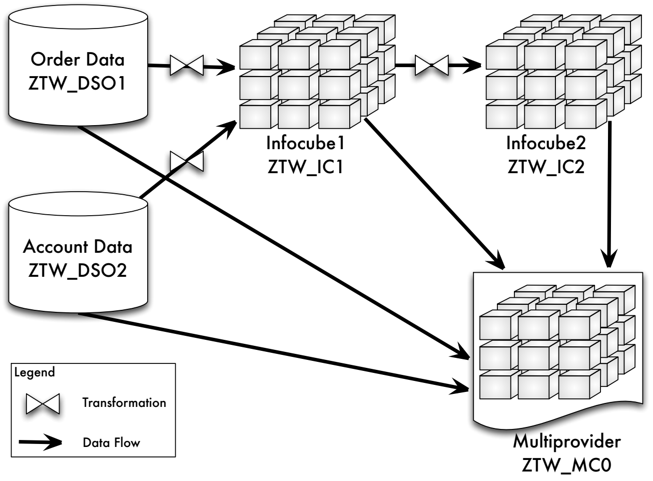

Fig. 5. Data Flow in the Case Study Example

In the following, we briefly describe the system terms and explain particular possibilities to obtain the technical measurements for our proposed areas data supply, data actualization, data reorganization, and cost. We refer for further detailed information regarding our used system to the SAP help portal [37].

The SAP BW statistics [38] include information regarding statistics on data loads and data status as well as statistics regarding query execution times. Note, measurements related to run time, data propagation, and data processing are stored within the system in principle.

A detailed description of available business warehouse statistics and information on technical content are available at [38]. In the following, we define the statistical data that we use for the multiprovider in our case study. For the data flow within our example please also see details in Fig. 5.

In the remainder of this paper, we use the term information provider for operational data stores (DSO), information cubes (IC), and multiprovider. Information cubes can be interpreted as data marts in a common DW architecture.

The multiprovider ZTW_MC0 provides aggregated data of frontend and query run time statistics. These data cover the area data supply with details on duration and frequency on data allocation.

Our case study is realized in SAP NetWeaver Business Warehouse, Release 7.40, with SAP HANA (HDB, Release 1.0) as database management system. Within the domain of our case study, we use order and account data that are extracted, transformed, and loaded into the DW at a scheduled time basis. Fig. 5 shows all warehouse objects including the data flow between the objects. All incoming data (after the ETL processes that extract the data from operational systems) are directly stored persistently in the order data store (ZTW_DSO1) and the invoice data store (ZTW_DSO2). Both data stores provide data that have to be transformed for Infocube 1 (ZTW_IC1). After another transformation, data are stored persistently in Infocube 2 (ZTW_IC2). Data in the data stores are homogenized, integrated, and adjusted in such a way that transformations regarding both infocubes do not change these data.

We define the transformations as follows:

- τ_{D1}: input data → ZTW_DSO1
- τ_{D2}: input data → ZTW_DSO2
- τ_{1I1}: ZTW_DSO1 → ZTW_IC1
- τ_{2I1}: ZTW_DSO2 → ZTW_IC1
- τ_{I2}: ZTW_IC1 → ZTW_IC2

Data actualization, which means data supply for all information providers respecting new entries is performed every two hours. Data reorganization is applied on a weekly basis. We do not include the initial loading of data into our case study. All information provider are directly connected to the multiprovider ZTW_MC0. This multiprovider generates all reports and enables further data analytics for our case study. However, these results are not persistently stored. We consider a group of reports r in our case study that consists of six reports $r_i \in R \quad i = \{1, \ldots, 6\}$. All reports r_i can be generated regarding the corresponding information provider ZTW_DSO1, ZTW_DSO2, ZTW_IC1, and ZTW_IC2.

We define the processing model within the business warehouse as follows: data supply (A) starts every uneven hour (starting at 1am), reports (B) are invoked in the business hours (from 8:00 to 18:00) and data reorganization (R) is scheduled to the weekend. The weekly sequence of processes is defined by:

$$5 \cdot [4 \cdot A - 2 \cdot B - A - 4 \cdot B - A - 4 \cdot B - A - 4 \cdot B - A - 4 \cdot B - A - 2 \cdot B - 3 \cdot A] - R.$$

The initial data assets for input data consist of 1.2 million data tuples for order and invoice data. These data are stored within the information providers. We assume that about 10.000 tuples are added on a weekly basis. The data detail level in both operational data stores are bills and receipts and position respectively. These data are stored in Infocube 1 without loss of information, i.e., ZTW_IC1 contains the same characteristics and indicators as both operational data stores. However, the last transformation to Infocube 2 reduces the information, which means that ZTW_IC2 does not contain all facts compared to both sources.

Table V gives an overview on all info objects (“X”) that are stored in the corresponding information provider. Note, we also classify this information with respect to measurement (Type M) or fact (Type F).

B. Measuring Required Decision Indicators

In this section, we present the different steps toward a decision on persistence by using our model. We investigate the question of an efficient data distribution scheme. Therefore, we decide whether or not both infocubes are required. This leads to the following three alternatives respecting the processing model described before:

- I: $\tau_{D1}/\tau_{D2} + ZTW\_D1/ZTW\_D2 \rightarrow \tau_{1I1}/\tau_{2I1} + ZTW_IC1 \rightarrow \tau_{I2} + ZTW\_IC2 \rightarrow \tau_R + R$
- II: $\tau_{D1}/\tau_{D2} + ZTW\_D1/ZTW\_D2 \rightarrow \tau_{1I1}/\tau_{2I1} + ZTW_IC1 \rightarrow \tau_R + R$
- III: $\tau_{D1}/\tau_{D2} + ZTW\_D1/ZTW\_D2 \rightarrow \tau_R + R$

Data supply: For evaluation of the data supply, we consider the execution times of six different report r_i that are daily invoked about 20 to 40 times within a five-day-week. For our comparison of the three different alternatives, we triplicate these reports and name them at the end with respect to

TABLE V. INFORMATION OBJECTS IN THE INFORMATION PROVIDER

InfoObject	Type	DSO1	DSO2	IC1	IC2
Order reason	M	X		X	
Order no.	M	X		X	
Account no.	M		X	X	
Calendar day	M	X	X	X	X
Calendar year / month	M	X	X	X	X
Classification of customer	M	X	X	X	X
Customer no.	M	X	X	X	X
Region ID	M	X	X	X	X
Material type	M	X	X	X	X
Material group	M	X	X	X	X
Material no.	M	X	X	X	X
Quantity unit	M	X	X	X	X
Employee no.	M	X	X	X	X
Position no.	M	X	X	X	
Product hierarchy	M	X	X	X	X
Category	M	X	X	X	X
City	M	X	X	X	X
Sales organization	M	X	X	X	X
Distribution channel	M	X	X	X	X
Currency key	M	X	X	X	X
Number of line items	F	X	X	X	X
Order quantity	F	X		X	X
Order value	F	X		X	X
Account quantity	F		X	X	X
Account value	F		X	X	X
VAT	F		X	X	X
VAT (in %)	F		X	X	X
Open order quantity	F	X		X	X
Open order value	F	X		X	X
Returns	F	X		X	X
Value of returns	F	X		X	X
Unit price	F	X	X	X	X

their alternative (I: ZTW_DSO1 + ZTW_DSO2, II: ZTW_IC1, III: ZTW_IC2). The initial detail level of report information besides measures and facts from Table V include the following facts:

- ZTW_Q01I/II/III: Sales organization - distribution channel - Category - Customer no. - Material no. - Calendar year / month
- ZTW_Q02I/II/III: Sales organization - distribution channel - Category - Employee no. - Calendar year / month
- ZTW_Q03I/II/III: Sales organization - City - Classification of customer - Customer no. - Calendar year / month
- ZTW_Q04I/II/III: Sales organization - Product hierarchy - Material group - Material type - Material no. - Calendar year / month
- ZTW_Q05I/II/III: City - Classification of customer - Customer no. - Calendar day
- ZTW_Q06I/II/III: Product hierarchy - Material group - Material type - Material no. - Calendar day

We use the multiprovider for data provision of these reports. The defined analysis determines all required evaluation indicators (average response times and frequencies). Additionally, this is done for every alternative and report. We use the multiprovider 0TCT_MC01 for data provision of these reports. The defined analysis determines all required evaluation indicators (average response times and frequencies). Additionally, this is done for every alternative and report, compare Table VI.

Data actualization: Every two hours, new operational data are loaded into the BDW according to the data flow in Fig. 5. This means that data actualization takes place twelve times every day. In our evaluation, all data transfer processes (τ) from source objects to targets information objects are included. For both operational data stores, we also include data activation time for integrating new data into the stores, see for more details [39], [37]. We consider the complete duration from input data to the last stage of persistence. For simplicity, we do not consider parallel execution. For all alternatives we have to evaluate the following processes:

- I: data transfer: $\tau_{D1}, \tau_{D2}, \tau_{1I1}, \tau_{2I1}, \tau_{I2}$
  data activation: ZTW_DSO1, ZTW_DSO2
- II: data transfer: $\tau_{D1}, \tau_{D2}, \tau_{1I1}, \tau_{2I1}$
  data activation: ZTW_DSO1, ZTW_DSO2
- III: data transfer: τ_{D1}, τ_{D2}
  data activation: ZTW_DSO1, ZTW_DSO2

We use again the multiprovider for reporting the measurements. All indicators are grouped by information provider and processes and given in Table VII.

Data reorganization: The complete database is reorganized once a week. For the operational data stores this includes deletion from so-called change logs (CL) and for both infocubes data compression. For details see again [37]. We consider for data reorganization the following processes:

- I: CL deletion: ZTW_DSO1, ZTW_DSO2 compression: ZTW_IC1, ZTW_IC2
- I: CL deletion: ZTW_DSO1, ZTW_DSO2 compression: ZTW_IC1
- I: CL deletion: ZTW_DSO1, ZTW_DSO2

TABLE VII. DATA ACTUALIZATION AND REORGANIZATION

	Process type	Average time (s)	Frequency	Overall time (s)
Data actualization				
DSOs	Data transfer	7.972	120	956.633
	Data activation	2.701	120	324.116
	Overall	5.336	240	1280.749
IC1	Data transfer	7.981	120	957.727
IC2	Data transfer	7.199	60	431.916
Data reorganization				
DSOs	CL deletion	22.082	1	22.082
IC1	Compression	26.717	1	26.717
IC2	Compression	24.138	1	24.138

Our data on the measurements is again evaluated within the multiprovider and we present the results in Table VII. *Cost*: Cost is generated by work on the objects in the alternatives. This includes work on information providers as well as transformations and further processes. Note, cost cover all operations in Alternative I and therefore, operational data stores as well as both infocubes have to be considered.

For our three alternatives, this leads to:

- I: $\frac{\tau_{D1}}{\tau_{D2}} + \frac{ZTW\_DSO1}{ZTW\_DSO2} \rightarrow \frac{\tau_{1I1}}{\tau_{2I1}} + ZTW\_IC1 \rightarrow \tau_{I2} + ZTW\_IC2$
- II: $\frac{\tau_{D1}}{\tau_{D2}} + \frac{ZTW\_DSO1}{ZTW\_DSO2} \rightarrow \frac{\tau_{1I1}}{\tau_{2I1}} + ZTW\_IC1$
- III: $\frac{\tau_{D1}}{\tau_{D2}} + \frac{ZTW\_DSO1}{ZTW\_DSO2}$

The determination and assignment for all accrued expenses is quite complicate and can only be estimated in some ways. We determined the working cost at a time base. Our application is extended by one object, that depends on all other objects and therefore all data must be reloaded. Modeling time is 63 minutes, which we distribute equally on all involved seven objects. For our three alternatives, we assign the relevant cost per object, i.e., I: 63 minutes, II: 49 minutes, and III: 28 minutes. We assume that such a modeling is required twice a year in practice.

Maintenance requires 10 minutes for checking activities and restart of two aborted loading processes every week. Due to the fact that all alternatives are affected, we assign these cost to all alternatives. The cost for quality assurance is originated from data monitoring that is directly performed after the model

TABLE VI. MEASUREMENTS FOR DATA SUPPLY

	Report	Average response time (s)	Frequency	Overall time (s)	Time Data Manager (s)	Time OLAP (s)	Time Frontend (s)
Data Supply DSOs	ZTW_Q01A	4.780	100	478.077	26.504	447.678	3.893
	ZTW_Q02A	2.887	100	288.668	17.088	268.557	3.024
	ZTW_Q03A	2.812	100	281.231	19.236	258.701	3.293
	ZTW_Q04A	4.020	100	402.016	19.612	378.97	3.436
	ZTW_Q05A	1.794	200	358.738	18.439	333.891	6.418
	ZTW_Q06A	2.614	200	522.850	18.157	498.225	6.464
	Overall	2.915	800	2,331.58	119.036	2,186.022	26.528
Data Supply IC1	ZTW_Q01B	4.289	100	428.905	3.222	421.953	3.716
	ZTW_Q02B	2.638	100	263.829	3.089	257.778	2.959
	ZTW_Q03B	2.491	100	249.084	3.117	242.799	3.167
	ZTW_Q04B	3.697	100	369.728	3.196	363.104	3.430
	ZTW_Q05B	1.627	200	325.352	3.560	315.897	5.896
	ZTW_Q06B	2.431	200	486.171	2.995	477.012	6.153
	Overall	2.654	800	2,123.069	19.179	2,078.543	25.321
Data Supply IC2	ZTW_Q01C	4.248	100	424.771	3.147	417.939	3.681
	ZTW_Q02C	2.636	100	263.619	2.994	257.709	2.919
	ZTW_Q03C	2.465	100	246.497	3.076	240.245	3.172
	ZTW_Q04C	3.721	100	372.125	3.189	365.511	3.428
	ZTW_Q05C	1.620	200	324.063	4.131	314.060	5.867
	ZTW_Q06C	2.437	200	487.301	2.878	478.217	6.197
	Overall	2.648	800	2,118.376	19.415	2,073.681	25.264

TABLE VIII. EVALUATION INDICATORS FOR COST

Cost type	I	II	III
Modeling	2.42	1.88	1.08
Maintenance	10	10	10
Quality Assurance	2.31	1.73	1.15
$\sum$	14.73	13.61	12.23

change. For each object this took approximately 15 minutes, which results in I: 60 minutes, II: 45 minutes, III: 30 minutes. Again, this effort has to be taken twice a year. As period under consideration we use a week and give our estimates in minutes per alternative. We present our estimation results in Table VIII.

C. Applying our MCDA-Decision Approach

For evaluation of the above described measurements we use our model from Section IV. We compute the values according to Equations 1 to 10. We present the results in Table IX.

TABLE IX. UTILITY VALUES FOR ALTERNATIVES I, II, III

		Alternative		
Class		I	II	III
Data Supply	Overall in s	2118,380	2123,070	2331,580
	Weighted utility	0,344	0,343	0,313
Data actualization	Overall in s	2670,420	2238,480	1280,760
	Weighted utility	0,234	0,279	0,487
Data reorganization	Overall in s	72,937	48,799	22,082
	Weighted utility	0,172	0,258	0,570
Cost	Overall in min per week	14,731	13,615	12,231
	Weighted utility	0,304	0,329	0,366

The decision maker E_1 prefers a fast data supply and low cost at second place. We make a pairwise comparison between all four classes, where 9 is highest preference and 1 / 9 is lowest, 1 means equal. Note, for a sound comparison only one way comparisons are required, the other value is reciprocal. We present the preferences in Table X.

TABLE X. PREFERENCES OF DECISION MAKER E_1

	Data supply	Data actualization	Data reorganization	Cost
Data supply	1	7	9	5
Data actualization	1 / 7	1	1	1 / 5
Data reorganization	1 / 9	1	1	1 / 7
Cost	1 / 5	5	7	1

With the given preferences, we apply the MCDA approach for determining the corresponding weights for each class. This leads to the following intermediate results:

$$w_{Data\ Supply} = 0.65, w_{Data\ Actualization} = 0.06$$

$$w_{Data\ Reorganization} = 0.05, w_{Cost} = 0.24.$$

Applying Equation 10, the ranking of all alternatives for decision maker E_1 is derived as $III \succ II \succ I$. This means that including all indicators and user preferences of E_1, Alternative III is preferred and both information cubes 1 and 2 should be omitted.

D. Discussion

With our proposed decision methodology it is possible to decide which materialization objects should be considered in a BDW. Due to the fact that our approach enables an objective quantification of the technical and administrative indicators, we provide a sound decision basis for persistence even in the context of in-memory databases. Furthermore, it is also possible to include user preferences that weight the different classes in such a way that either one decision maker or even a group of decision makers identify the best solution of persistence level within the business data warehouse.

Because our measurements are already normalized, we can give a decision support without user preferences. This leads to a ranking of alternatives as: $III \succ II \succ I$ with values for utility function:

$$U(I) \approx 0.26, U(II) \approx 0.30, U(III) \approx 0.43.$$

Including the preferences of decision maker E_1 changes the values of the utility function to $U(I) \approx 0.32, U(II) \approx 0.33, U(III) \approx 0.35$. However, the ranking of alternatives is the same as before. Note, the differences in the utilities converges by including the specific preferences. Due to this convergence, it is advisable to reconsider the preferences and apply a sensitivity analysis to identify a possible change in the ranking.

VI. Related Work

Persistence of redundant data in DW systems is closely related to materialized views and their incremental maintenance, in conjunction with incremental loading of DW. In the selection of materialized views, all data assets should be included that require a long processing time or that are used frequently [40]. Within the DW, a consideration of the underlying data cube lattice is quite important. Dependencies and sparsely populated data areas can be efficiently included in these considerations and optimize the DW design in important facets. In literature, there are several works discussing choice and actualization of materialized views. The proposed algorithms use only cost-based approaches to identify which materialized views are optimal [41]. Query response times of optimal views are combined with the utility value, whereas Gupta [42] requires for a monotone utility function, see also [43].

In the domain of static models a monotone utility function is used and cost of actualization is disregarded. Shukla et al. [44] define a "benefit per unit space" and use it as optimization criterion. Due to np complexity of the search for an optimum, a Greedy approach is required [45]. Shukla et al. [46] enhance this approach to multi-cube models. Including actualization cost into the utility function results in a non-monotone function. Therefore, static models that consider only cost lead to suboptimal solutions. Gupta and Mumick [47] examine a comprehensive set of nodes and deconstruct the data cube lattice. Further works in this area regard actualization cost, too, for instance [48], [49], [50].

In the domain of dynamic models, a selection of summation data is the subjective. These models include also changes in the data consumptions (or analytics). Materialized views are checked whether they are deleted, temporarily stored, or stored in a more persistent way. Scheuermann et al. [51] and later Shim et al. [52] define the profit that is achieved by a materialization related to a query. However, none of these approaches include cost that result from user workload or even preferences of decision-makers in their models. For an overview on materialized views see [53]. A formal model on the view selection problem for data warehouses is presented in [54]. However, user preferences are not included in the model. [55] give a general framework for the view selection problem for designing a DW, but also for the evolution of data warehouses. For improving the design and integration of data into the DW, a focus on the ETL process is for instance set in [56].

Another important issue is updating materialized views in ETL processes. Jörg and Deßloch [57] remark that scientific examples and case studies are often far away from practice, which we address in this paper with our case study. Furthermore, incremental loading in the DW is comparable to incremental updates of materialized views. Nevertheless, the authors do not consider reasons for persistence within a DW.

VII. Conclusion and Outlook

Persistent data in BDW systems require effort for maintenance among others. In order to avoid dispensable data persistence, the need for such persistence has to be defined by the purpose of the data. Therefore, we classify reasons for persistence in such systems. Based on this, we come up to decide whether to store data or not. For mandatory stored data, this decision is clear. However, the need for essentially stored data is more difficult to decide and goes beyond pure cost-based comparisons of system data. We present an approach that combines system data with user workload. Preferences of decision-makers are also considered by including methods of MCDA to be able to support decision-making. In our opinion, this topic matters particularly with regard to the use of IMDB in BDW systems.

Our future work will extend the model to evaluate further scenarios, additional data cubes, including the definition of a proper set of formulas. We also include report variants that cannot be operated by a cube due to missing elements (cf. Section IV) . Moreover, we want to perform extensive tests, including further real-life data and additionally, apply and evaluate our model to different DW architectures. Finally, a support within the DW system as part of the optimization could enhance the system in a semi-automatic way.

References

[1] R. Winter. (2008, Apr) Why are data warehouses growing so fast? BeyeNETWORK. [Online]. Available: http://www.b-eye-network.com/print/7188

[2] A. Zeier, A. Bog, J. Schaffner, J. Krueger, and H. Plattner, "ETL-less zero redundancy system and method for reporting OLTP data," Mar. 26 2009, WO Patent App. PCT/EP2008/062,646.

[3] V. Belton and T. J. Stewart, *Multiple criteria decision analysis - an integrated approach.* Springer, 2002.

[4] T. Winsemann and V. Köppen, "Persistence in data warehousing," in *RCIS*. IEEE, 2012, pp. 445–446.

[5] T. Winsemann, V. Köppen, and G. Saake, "A layered architecture for enterprise data warehouse systems," in *Advanced Information Systems Engineering Workshops*, ser. Lecture Notes in Business Information Processing, M. Bajec and J. Eder, Eds., vol. 112. Springer, 2012, pp. 192–199, doi:10.1007/978-3-642-31069-0.

[6] B. A. Devlin and P. T. Murphy, "An architecture for a business and information system," *IBM Systems Journal*, vol. 27, no. 1, pp. 60–80, 1988.

[7] V. Poe, S. Brobst, and P. Klauer, *Building a Data Warehouse for Decision Support*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.

[8] V. Köppen, G. Saake, and K.-U. Sattler, *Data Warehouse Technologien*, 2nd ed. MITP, May 2014, in German.

[9] B. A. Devlin, "Business integrated insight (BI^2): Reinventing enterprise information management," 9sight Consulting, Tech. Rep., 2009.

[10] T. Ariyachandra and H. Watson, "Key organizational factors in data warehouse architecture selection," *Decision Support Systems*, vol. 49, no. 2, pp. 200 – 212, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923610000436

[11] J. Haupt, "The BW Layered Scalable Architecture (LSA)," SAP: Training Material, March 2009.

[12] T. Winsemann and V. Köppen, "Persistence in enterprise data warehouses," Otto-von-Guericke University Magdeburg, Technical Reports 2-2012, March 2012.

[13] G. P. Copeland and S. N. Khoshafian, "A decomposition storage model," in *Proceedings of the 1985 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '85. New York, NY, USA: ACM, 1985, pp. 268–279.

[14] M. J. Turner, R. Hammond, and P. Cotton, "A DBMS for large statistical databases," in *Proceedings of the Fifth International Conference on Very Large Data Bases - Volume 5*, ser. VLDB '79. VLDB Endowment, 1979, pp. 319–327.

[15] D. Ślęzak, J. Wróblewski, V. Eastwood, and P. Synak, "Brighthouse: An analytic data warehouse for ad-hoc queries," *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1337–1345, Aug. 2008.

[16] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. O'Neil, P. O'Neil, A. Rasin, N. Tran, and S. Zdonik, "C-store: A column-oriented DBMS," in *Proceedings of the 31st International Conference on Very Large Data Bases*, ser. VLDB '05. VLDB Endowment, 2005, pp. 553–564.

[17] D. Abadi, S. Madden, and M. Ferreira, "Integrating compression and execution in column-oriented database systems," in *SIGMOD*. New York, NY, USA: ACM, 2006, pp. 671–682.

[18] L. Henkes, "SAP NetWeaver BW 7.3 powered by HANA - Introduction," asug Annual conference, 2012.

[19] V. Murthy and P. Deshpande, "Oracle exalytics in-memory machine: A brief introduction," Oracle Corporation, Tech. Rep., 2014.

[20] H. Plattner, "A common database approach for OLTP and OLAP using an in-memory column database," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '09. New York, NY, USA: ACM, 2009, pp. 1–2.

[21] H. Plattner and A. Zeier, *In-Memory Data Management: Technology and Applications*, 2nd ed. Springer, 2012.

[22] O. Shmueli and A. Itai, "Maintenance of views," in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '84. New York, NY, USA: ACM, 1984, pp. 240–255.

[23] I. S. Mumick, "The rejuvenation of materialized views," in *Information Systems and Data Management*, ser. Lecture Notes in Computer Science, S. Bhalla, Ed., vol. 1006. Springer Berlin Heidelberg, 1995, pp. 258–264.

[24] Y. Zhuge, H. García-Molina, J. Hammer, and J. Widom, "View maintenance in a warehousing environment," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '95. New York, NY, USA: ACM, 1995, pp. 316–327.

[25] M. Staudt and M. Jarke, "Incremental maintenance of externally materialized views," in *Proceedings of the 22th VLDB*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, pp. 75–86.

[26] N. Roussopoulos, "Materialized views and data warehouses," *SIGMOD Rec.*, vol. 27, no. 1, pp. 21–26, Mar. 1998.

[27] M. Schäler, A. Grebhahn, R. Schröter, S. Schulze, V. Köppen, and G. Saake, "Queval: Beyond high-dimensional indexing à la carte," *PVLDB*, vol. 6, no. 14, pp. 1654–1665, SEP 2013.

[28] V. Köppen, M. Schäler, and R. Schröter, "Toward variability management to tailor high dimensional index implementations," in *RCIS*. IEEE, 2014, pp. 452–457.

[29] E. Triantaphyllou, *Multi-criteria Decision Making Methods: A Comparative Study*, ser. Applied Optimization. Dordrecht: Kluwer Academic Publisher, 2000, vol. 44.

[30] T. L. Saaty, "Decision making with the analytic hierarchy process," *Scentia Iranica*, vol. 9, no. 3, pp. 215–229, 2002.

[31] B. Berendt and V. Köppen, "Improving ranking by respecting the multidimensionality and uncertainty of user preferences," in *Intelligent Information Access*, ser. Studies in Computational Intelligence, vol. 301. Springer Berlin / Heidelberg, 2010, pp. 39–56.

[32] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.

[33] T. L. Saaty, *The analytic hierarchy process : planning, priority setting, resource allocation*. New York: McGraw-Hill, 1980.

[34] ——, "Decision making with the analytic hierarchy process," *Int. J. Services Sciences*, vol. 1, no. 1, pp. 83–98, 2008.

[35] T. Winsemann, "Bewertung von Datenpersistenz in Business-Data-Warehouse-Systemen mithilfe multikriterieller Entscheidungsmodelle," Ph.D. dissertation, Otto-von-Guericke-University Magdeburg, 2015, in German.

[36] *TPC Benchmark DS - Standard Specification Version 1.1.0*, Transaction Processing Perfromance Council (TPC) Std., April 2012. [Online]. Available: www.tpc.org

[37] SAP AG, "SAP Business Warehouse," 2014. [Online]. Available: http://help.sap.com/nwbw?current=nw74

[38] ——, "BW Statistics," 2015. [Online]. Available: http://help.sap.com/saphelp_nw73/helpdata/en/44/3521c7bae848a1e10000000a114a6b/content.htm

[39] A. Palekar, B. Patel, and S. Shiralkar, *SAP NetWeaver BW 7.3 – Practical Guide*, 2nd ed. Galileo Press, 2013.

[40] N. Pendse, "The FASMI definition for OLAP," Business Intelligence, August 1995.

[41] T. L. Achs, "Optimierung der materialisierten sichten in einem datawarehouse auf der grundlage der aus einem erp-system übernommenen operativen daten," Ph.D. dissertation, WU Vienna University of Economics and Business, 2004, in German.

[42] H. Gupta, "Selection of views to materialize in a data warehouse," in *ICDT '97*, ser. ICDT '97. London: Springer, 1997, pp. 98–112.

[43] H. Gupta and I. S. Mumick, "Selection of views to materialize in a data warehouse," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 24–43, 2005.

[44] A. Shukla, P. Deshpande, and J. F. Naughton, "Materialized view selection for multidimensional datasets," in *Proceedings of the 24th VLDB*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 488–499.

[45] V. Harinarayan, A. Rajaraman, and J. D. Ullman, "Implementing data cubes efficiently," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '96. New York, NY, USA: ACM, 1996, pp. 205–216.

[46] A. Shukla, P. M. Deshpande, and J. F. Naughton, "Materialized view selection for multi-cube data models," in *Advances in Database Technology — EDBT 2000*, ser. Lecture Notes in Computer Science, C. Zaniolo, P. Lockemann, M. Scholl, and T. Grust, Eds., vol. 1777. Springer Berlin Heidelberg, 2000, pp. 269–284.

[47] H. Gupta and I. S. Mumick, "Selection of views to materialize under a maintenance cost constraint," in *ICDT'99*. Springer, 1999, pp. 453–470.

[48] C. I. Ezeife, "Accommodating dimension hierarchies in a data warehouse view/index selection scheme," in *Systems Development Methods for the Next Century*, W. Wojtkowski, W. Wojtkowski, S. Wrycza, and J. Zupančič, Eds. Springer US, 1997, pp. 195–211.

[49] D. Theodoratos and T. K. Sellis, "Data warehouse configuration," in *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB'97)*, M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, Eds. Athens, Greece: Morgan Kaufmann, 1997, pp. 126–135.

[50] ——, "Data warehouse schema and instance design," in *Conceptual Modeling ER 98*, ser. Lecture Notes in Computer Science, T.-W. Ling, S. Ram, and M. Li Lee, Eds., vol. 1507. Springer Berlin Heidelberg, 1998, pp. 363–376.

[51] P. Scheuermann, J. Shim, and R. Vingralek, "Watchman: A data warehouse intelligent cache manager," in *Proceedings of the 22th International Conference on Very Large Data Bases*, ser. VLDB '96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, pp. 51–62.

[52] J. Shim, P. Scheuermann, and R. Vingralek, "Dynamic caching of query results for decision support systems," in *Proceedings of the 11th International Conference on Scientific and Statistical Database Management*, ser. SSDBM '99. Washington, DC, USA: IEEE Computer Society, 1999, p. 254.

[53] R. Chirkova and J. Yang, "Materialized views," *Foundations and Trends in Databases*, vol. 4, no. 4, pp. 295–405, 2011.

[54] R. Chirkova, A. Y. Halevy, and D. Suciu, "A formal perspective on the view selection problem," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 11, no. 3, pp. 216–237, 2002.

[55] D. Theodoratos and M. Bouzeghoub, "A general framework for the view selection problem for data warehouse design and evolution," in *Proceedings of the 3rd ACM International Workshop on Data Warehousing and OLAP*, ser. DOLAP '00. New York, NY, USA: ACM, 2000, pp. 1–8.

[56] A. Simitsis, K. Wilkinson, M. Castellanos, and U. Dayal, "Qox-driven ETL design: reducing the cost of ETL consulting engagements," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009, pp. 953–960.

[57] T. Jörg and S. Deßloch, "Towards generating ETL processes for incremental loading," in *Proceedings of the 2008 International Symposium on Database Engineering & Applications*, ser. IDEAS '08. New York, NY, USA: ACM, 2008, pp. 101–110.