Quality metrics Validation in View Maintenance Models of Data Warehouse

Anjana Gosain
University School of Information Technology
GGS Indraprastha University
Delhi, India

Sangeeta Sabharwal
Netaji Subhas Institute of Technology
Delhi University
Delhi, India

Rolly Gupta
Netaji Subhas Institute of Technology
Delhi University
Delhi, India

ABSTRACT
Data warehouses are huge repositories designed to enable the knowledge workers to make better and faster decisions. Due to its significance in strategic decision making, there is a need to assure data warehouse quality in the presence of evolution events which may be generated as result of change in schema / software or data warehouse requirements. One of the factors affecting the data warehouse quality is view maintenance models quality. Although there are some useful guidelines for designing good view maintenance models, but objective indicators, i.e., metrics are needed to help designers to develop quality view maintenance models. In our previous work, a quality metric for View maintenance models of data warehouse is proposed [25] However, the proposal overall lacks theoretical and empirical validation of the metric proposed. Hence, the metric practical utility could not be established. This paper validates the metrics both theoretically and empirically. The theoretical validation is performed using Zuse framework [7] while empirical validation is carried out using MVPP (Multiple View Processing Plan) to explore the relationship between the proposed metrics and cost efficiency of View maintenance models. The results show that all the four metrics NBR, NVM, NAMV and NFMV have significant impact on the cost efficiency of View maintenance models.

General Terms
Zuse framework, MVPP, Validation.

Keywords
Data Warehouse, Data Warehouse Evolution, View maintenance models.

1. INTRODUCTION
Having accurate and up-to-date data warehouses is essential for Business Intelligence and Decision Support. A data warehouse (DW) design, apart from performance guarantees, should also provide correctness guarantees. Every time an evolution event occurs anywhere in the warehouse environment (e.g., a design change at the operational sources) it should be smoothly absorbed without causing any further inconvenience. For achieving this, the warehouse and its counterparts should be easily maintainable and the process of populating it should not be destructed by evolution events.

The Evolution of a DW constitute the backbone of a typical data warehouse architecture. Most of the research for improving DW evolution designs has focused solely on improving performance. However, based on practical experience, maintenance makes up for up to 60 % of the resources spent in a warehouse project [34], and therefore, maintainability is an important factor for the determination of the quality of a design [19,36]. Although practitioners are well aware of this problem, still, we miss a formal and concrete answer to deals with the quality perspective for View Maintenance models of DW. George et.al. [49] proposed a set of metrics for the evaluation of the vulnerability warehouse modules to future changes and for the assessment of the quality of alternative designs of the warehouse. These proposed metrics are based on graph-theoretic properties of the warehouse graph to assess the sensitivity of the graph to a set of possible events.

Most of the researches consider only structural properties of the DW evolution or constructs internal to the underlying databases. But the employed approach neither accounts for the constructs surrounding the database into their models, nor the fact that a software construct evolves over time. In practice, the problem is hard since changes in the schema of database-centric systems affect not only its internals but also the surrounding deployed applications. Hence, the minimal interdependence of these software modules results in higher tolerance to subsequent changes and should be measured with a principled theory. Related work for evolution of data-intensive applications [9], view redefinition [10,17,26], and data warehouse evolution [4,5,11,14] has provided rewriting techniques and theoretical cost models.

Related work [30] includes an approach to impact analysis and management of schema evolution, which represents the structural properties of the data warehouse schema, along with any views and queries defined over this schema, as a graph. Graph-based model [49] captures all the parts (or, modules) of an environment, i.e., relations, views, and queries (which are practically the parts of ETL scripts that work the underlying data, or the elementary activities of a GUI based scenario that are involved in the ETL process). In [59] authors focused on a set of graph-theoretic metrics for the prediction of evolution impact and their fitness into real-world ETL scenarios. However, these research works are dealing at design level only without considering the maintenance and quality perspective for VM models of DW.

Following this consideration, we have defined a set of metrics for the VM models of DW [25]. In this paper, we built upon the aforementioned approaches with the goal of theoretically and empirically validating the proposed metrics. We formally present these metrics and show that such metrics typically act as predictors for the vulnerability of a VM module of a data warehouse (e.g., a dimension table, or external, an aggregated measure etc.) to future changes to the structure of the warehouse. Secondly, they facilitate the assessment of the quality of alternative designs of the data warehouse VM models with a particular viewpoint on the evolution of the data warehouse.

Theoretical validation assures that the metric is correctly defined and the metric actually measures what it purports to
2. PROPOSED METRICS

Metrics should be defined on the basis of clear measurement goals. So, we have defined a set of metrics for the VM models of DW. The proposed sets of metrics are dealing with two major characteristics. Firstly, they act as predictors for the vulnerability of a VM module of a data warehouse (e.g., a dimension table, or external, an aggregated measure etc.) to future changes to the structure of the warehouse. Secondly, they facilitate the assessment of the quality of alternative designs of the data warehouse VM models with a particular viewpoint on the evolution of the data warehouse. The proposed metrics have been defined for measuring the cost efficiency of data warehouse VM models [25].

1. Number of Base relations: These metric counts the number of base Relations in the view maintenance models for data warehouse.

2. Number of Views materialized: These metric counts the Number of Views Materialized in the view maintenance models for data warehouse.

3. Number of Attributes in materialized views: This metric records the total number of attributes considered for data warehouse VMM.

4. Number of Foreign Keys in materialized views: This metric counts the number of Foreign keys in data warehouse VMM.

3. THEORETICAL VALIDATION OF METRICS USING ZUSE FRAMEWORK

Zuse (1998) describes measurement as a detour. Measurement is more than producing numbers, it is the combination of empirical entities with numerical entities. These empirical relations can be denoted with the symbols “•>” and “• >=”, respectively. An empirical relational system is a triple: 

\[ (A, \cdot >, \cdot >=, o) \]

where \( A \) is a non-empty set of objects, \( \cdot > \) is an empirical relation on \( A \), and \( o \) is a closed binary (concatenation) operation on \( A \). In many cases we are not able to produce directly relevant empirical results due to the difficulty of the question we deal with. With the aid of mathematics and statistics “this intelligence barrier” can be overcome: the empirical objects and relationships are mapped into proper numerical objects and relationships. A numerical relational system can be defined as \( B = (R, >, +) \), where \( R \) are the real numbers, \( > \) is a relation on \( R \), and + a closed binary operation on \( R \). A measure is then a mapping \( u : A \rightarrow R \) such that

\[ a \cdot b = b \Leftrightarrow u(a) > u(b); \quad \forall a, b \in A \]

Once the mapping is established, mathematics and statistics can then be used to process the information (e.g., working out means or variances). Measurement theory also leads to conditions where numerical statements can be translated back into empirical statements. To check whether the measure satisfies the users needs, Zuse proposes an internal validation, based on the comparison between the empirical interpretation of numbers and the empirical statements in the real world. The combination rule must be defined as:

\[ U(A_1A_2) = f(u(A_1), u(A_2)) \]

where \( A_1, A_2, A_1A_2 \in A \) and \( f(A_1A_2): A \times A \rightarrow A \). This concatenation operation \( (\cdot) \) can be contra-intuitive in the area of software engineering because it is not necessary to combine objects in reality. However, it provides a means for building up complex measurement structures, giving a more precise interpretation of numbers.

On this framework, Zuse defines a set of axioms for measures which gives rise to distinct structures. In Table 1, we present the most important ones. In software measurement, there are sufficient frameworks with the next five scale types that are defined by admissible transformations. They are, in hierarchical order: nominal, ordinal, interval, ratio and absolute. Each scale type is defined by admissible transformations. Software measurement starts with the ordinal scale. Measures may be classified in a scale type, depending on whether or not they assume an extensive structure. When a measure accomplishes this structure, it also accomplishes the independence conditions and can be used on the ratio scale levels. If a measure does not satisfy the modified extensive structure, the combination rule will exist or not depending on the independence conditions. When the independence conditions is assumed, but the modified extensive structure is not assumed, such type of scale is the ordinal scale.
Table 1: Zuse Framework [7]

<table>
<thead>
<tr>
<th>Modified extensive structure</th>
<th>Independence conditions</th>
<th>Modified relation of belief</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axiom 1: ( (A, \bullet \geq) ) (weak order)</td>
<td>C1: ( A_1 \rightarrow A_2 \Rightarrow A_1 \geq A_2 )</td>
<td>MRB1: ( A, B \in \mathcal{I}; A \geq B ) or ( B \geq A ) (completeness)</td>
</tr>
<tr>
<td>Axiom 2: ( A_1 \circ A_2 \bullet \geq A_1 ) (positivity)</td>
<td>C2: ( A_1 \rightarrow A_2 \Rightarrow A_1 \geq A_2 )</td>
<td>MRB2: ( A, B, C \in \mathcal{I}; A \geq B ) and ( B \geq C \Rightarrow A \geq C ) (transitivity)</td>
</tr>
<tr>
<td>Axiom 3: ( A_1 \circ (A_2 \circ A_3) ) (weak associativity)</td>
<td>C3: ( A_1 \bullet \geq A_2 \Rightarrow A_1 \rightarrow A_2 )</td>
<td>MRB3: ( A \subseteq B \Rightarrow A \geq B ) (dominance axiom)</td>
</tr>
<tr>
<td>Axiom 4: ( A_1 \circ A_2 = A_2 \circ A_1 ) (weak commutativity)</td>
<td>C4: ( A_1 \bullet \geq A_2 \Rightarrow A_1 \rightarrow A_2 )</td>
<td>MRB4: ( (A \geq B, A \cap C = \emptyset) \Rightarrow (A \geq B = A \cup C \bullet B \cup C) ) (partial monotonicity)</td>
</tr>
<tr>
<td>Axiom 5: ( A_1 \bullet \geq A_2 \Rightarrow A_1 \circ A ) (weak monotonicity)</td>
<td>C5: ( A_1 \bullet \geq A_2 \Rightarrow A_1 \rightarrow A_2 ) and ( A_1 \bullet \geq A_2 )</td>
<td>MRB5: ( A \in \mathcal{I}; A \bullet \geq 0 ) (positivity)</td>
</tr>
<tr>
<td>Axiom 6: If ( A_3 \bullet \geq A_4 ) then for any ( A_1, A_2 ), then there exists a natural number ( n ) such that ( A_1 \circ n. A_3 \bullet \geq A_2 \circ n. A_4 ) (Archimedean axiom)</td>
<td>As we know, binary relation ( \bullet \geq ) is called weak order if it is transitive and complete: ( A_1 \circ A_2 \bullet \geq A_3 \iff A_1 \rightarrow A_2 \iff A_3 \Rightarrow A_4 ) and ( A_1 \bullet \geq A_2 ) and ( A_2 \bullet \geq A_1 )</td>
<td>Natural join is defined generally as (Elmasri and Navathe (1999)): ( Q \rightarrow T_{(\text{list1})} \circ (\text{list2}) )</td>
</tr>
</tbody>
</table>

In the following section, we adapt this framework to databases to verify the fulfillment of the axioms for the metrics proposed.

3.1 Characterization of metrics

In relational database systems, and for our purposes, the empirical relational system could be defined as \( T = (T, \bullet \geq, \circ) \) where \( T \) is a non-empty set of relations (tables), \( \bullet \geq \) is the empirical relation “more or equal complex than” on \( T \), while \( \circ \) is a closed binary (concatenation) operation on \( T \). In our case we will choose “natural join” as the concatenation operation.

1. Number of attributes in materialized views:

The NA measure is a mapping: \( NA : T \rightarrow \mathbb{R} \) such that the following holds for all relations \( T_i \) and \( T_j \in T : T_i \bullet \geq T_j \Rightarrow NA(T_i) \geq NA(T_j) \).

So, the combination rule for NA can be defined as:

\[ NA(T_i \circ T_j) = NA(T_j) + NA(T_j) - NA(T_i \cap T_j) \]

where \( NA(T_i \cap T_j) \) is the number of attributes which are common to (belong to the intersection of) \( T_i \) and \( T_j \).

Making the formal verification, NA can be characterized as a measure above the level of the ordinal scale, assuming the modified relation of belief.

2. Number of Foreign Key in materialized views:

The RD measure is a mapping: \( RD : T \rightarrow \mathbb{R} \) such that the following holds for all relations \( T_i \) and \( T_j \in T : T_i \bullet \geq T_j \Rightarrow RD(T_i) \geq RD(T_j) \).

In order to obtain the combination rule for RD, we have to consider if the concatenation (by natural join) between tables is made by foreign key, i.e. the number of foreign keys affected (decreasing in one), and not affected in other cases.

So, we can characterize the combination rule for RD as

\[ RD(T_i \circ T_j) = RD(T_j) + RD(T_j) - v \]

The formal validation, therefore is analogous to that previously presented. So, in summary, we can characterize RD as a measure above the level of the ordinal scale, assuming the modified relation of belief.
3. Number of Views Materialized:

The NV measure is a mapping: NV : T → R such that the following holds for all relations Ti and Tj ∈ T : Ti ⋈ Tj ⇒ NV(Ti) ≥ NV(Tj).

So, the combination rule for NV can be defined as:

\[ NV(T_i \cap T_j) = NV(T_i) + NV(T_j) - NV(T_i \cap T_j) \]

where NV(T_i \cap T_j) is the number of attributes which are common to (belong to the intersection of) Ti and Tj.

Making the formal verification, NV can be characterized as a measure above the level of the ordinal scale, assuming the modified relation of belief.

4. Number of Base Relations:

The BR measure is a mapping: BR : T → R such that the following holds for all relations Ti and Tj ∈ T : Ti ⋈ Tj ⇒ BR(Ti) ≥ BR(Tj).

So, the combination rule for BR can be defined as:

\[ BR(T_i \cap T_j) = BR(T_i) + BR(T_j) \]

Making the formal verification, BR can be characterized as a measure above the level of the ordinal scale, assuming the modified relation of belief.

4. EMPIRICAL VALIDATION

After proving theoretical validity of metrics defined in Section 3, in this section, we intend to corroborate that these metrics are really related to cost efficiency of View Maintenance Models of Data Warehouse, i.e. measure of internal consistency will be applied. This is done on the basis of empirical validation. We have conducted controlled experiment for this reason. The dependent variable in this study is cost efficiency and the independent variables are the View Maintenance metrics defined in Section 3.

Twenty one multidimensional schemas were collected to carry out this controlled experiment. The domains of these selected models were general and well known in order to avoid the problems arising out of domain understanding. The metrics values for each schema were collected manually using MVPP approach [11]. There is no subjectivity in calculation of metrics values, as the metrics considered in this work are calculated by simply counting (e.g. NBR is calculated by counting the base relations present in the schemas). The collected data was shown in Table 2.

<table>
<thead>
<tr>
<th>NBR</th>
<th>NVM</th>
<th>NAMV</th>
<th>NFMV</th>
<th>Cost (Query processing and Maintenance cost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>13</td>
<td>46</td>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>24</td>
<td>13</td>
<td>46</td>
<td>2</td>
<td>75</td>
</tr>
<tr>
<td>31</td>
<td>9</td>
<td>47</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td>27</td>
<td>5</td>
<td>35</td>
<td>6</td>
<td>63</td>
</tr>
<tr>
<td>18</td>
<td>7</td>
<td>30</td>
<td>3</td>
<td>41</td>
</tr>
<tr>
<td>24</td>
<td>7</td>
<td>37</td>
<td>5</td>
<td>54</td>
</tr>
<tr>
<td>40</td>
<td>8</td>
<td>54</td>
<td>7</td>
<td>91</td>
</tr>
<tr>
<td>46</td>
<td>9</td>
<td>64</td>
<td>9</td>
<td>95</td>
</tr>
<tr>
<td>22</td>
<td>9</td>
<td>38</td>
<td>5</td>
<td>56</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>28</td>
<td>2</td>
<td>35</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>23</td>
<td>1</td>
<td>37</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>25</td>
<td>1</td>
<td>39</td>
</tr>
<tr>
<td>19</td>
<td>9</td>
<td>34</td>
<td>3</td>
<td>45</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>42</td>
<td>4</td>
<td>66</td>
</tr>
<tr>
<td>26</td>
<td>13</td>
<td>49</td>
<td>3</td>
<td>62</td>
</tr>
</tbody>
</table>
The points cluster together, a trend can be discerned. To evaluate the cost, John Wiley, empirically validated with the help of case studies or carried out to generalise the results. These metrics need to be perspectives. Replicated studies with more comprehensive and other consecutive research could further complete this set by defining new metrics from other different quality attribute. This set of metrics may not already proposed efficiency of VM models but these metrics along with other contributing towards the cost efficiency of the VM models.

Although, this is not a complete set of metrics to assess cost may act as objective indicators for the industrial data and by involving professionals to draw strong conclusion which can be applied in practice.

6. REFERENCES

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>13</td>
<td>25</td>
<td>2</td>
<td>44</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>38</td>
<td>3</td>
<td>43</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>35</td>
<td>4</td>
<td>48</td>
</tr>
<tr>
<td>27</td>
<td>11</td>
<td>49</td>
<td>3</td>
<td>64</td>
</tr>
<tr>
<td>38</td>
<td>12</td>
<td>60</td>
<td>6</td>
<td>88</td>
</tr>
</tbody>
</table>

A scatterplot is plotted between the dependent and independent variables. It is aimed to display a relationship between two sets of variables. Depending on the fact, that how tightly the points cluster together, a trend can be discerned.

![Figure 1: Scatterplot](image)

In figure 1, the variable data points make a straight line, starting from near the origin up to the high y-values. This shows that there exists a positive correlation between the set of variables. The data points are closer enough, to make a line, showing higher correlation. It is therefore interpreted that there is statistically significant correlation between the metrics and the cost efficiency parameter of the data warehouse View Maintenance models. So, it is necessary to investigate the quantitative impact of View Maintenance metrics on the dependent variable.

5. CONCLUSION AND FUTURE WORK
In this work, we validated the metrics, to evaluate the cost efficiency of VM models for data warehouse. These metrics are theoretically validated using Zuse’s framework [7] and it is concluded that these metrics are theoretically sound and are characterised as either size or length measure. We have also conducted controlled experiment in order to provide the empirical validation of the proposed metrics. The empirical validation using controlled experiments suggests that the metrics have significant effect on the cost efficiency parameter.

Hence, this study shows that these metrics are significantly contributing towards the cost efficiency of the VM models. Although, this is not a complete set of metrics to assess cost efficiency of VM models but these metrics along with other already proposed may act as objective indicators for the quality attribute. This set of metrics may not be comprehensive and other consecutive research could further complete this set by defining new metrics from other different perspectives. Replicated studies with more data need to be carried out to generalise the results. These metrics need to be empirically validated with the help of case studies or when plotted, the closer is the data points, coming to form a straight line, the higher is the correlation between the two set of variables, or their exist a stronger relationship.
Technology and engineering, e-Publisher: CiteSeerX , 2009.


[38] Dragan Sahpaski, Goran Velinov, Boro Jakimovski, Margita Kon-Popovska,” Dynamic Evolution and Improvement of Data Warehouse Design” Fourth Balkan Conference in Informatics, IEEE,2009


[41] Xin Zhang, Elke A. Rundensteiner,” The SDCC Framework For Integrating Existing Algorithms for...


