A Taxonomy of Inaccurate Summaries and Their Management in OLAP Systems

John Horner and Il-Yeol Song

College of Information Science and Technology, Drexel University,
Philadelphia, PA 19104
(jh38, song)@drexel.edu

Abstract. Accurate summarizability is an important property in OLAP systems because inaccurate summaries can result in poor decisions. Furthermore, it is important to understand and identify the potential sources of inaccurate summaries. In this paper, we present a taxonomy of inaccurate summary factors and practical rules for handling them. We consolidate relevant terms and concepts in statistical databases with those in OLAP systems and explore factors that are important for measuring the impact of erroneous summaries. We discuss these issues from the perspectives of schema, data, and computation. This paper contributes to a comprehensive understanding of summarizability and its impact on decision-making. Our work could help designers and users of OLAP systems reduce unnecessary constraints caused by imposing rules to eliminate all summarizability violations and give designers a means to prioritize problems.

1 Introduction

Data warehouses contain large sets of subject oriented, integrated, historical, and relatively static data used for strategic decision making. Because data warehouses are typically magnitudes larger than operational systems, they typically contain many aggregate summaries of base data. Thus, accurate summaries are necessary to ensure that the decisions based on them are sound.

Summarizability refers to the property of whether performing an aggregate operation will result in an accurate result. Martyn [1] describes three design criteria necessary for all database systems, consisting of correctness, efficiency, and usability, and argues that data correctness is of utmost importance. Lenz and Shoshani [2] also argue that summarizability in online analytic processing (OLAP) is an important property because violating this condition can result in erroneous conclusions and decisions. Shoshani [3] further argues that it is important for OLAP systems to borrow some areas of statistical database systems, such as strict classification hierarchies and the distinction between summary data from attributes.

Prior research in both statistical databases and OLAP has explored numerous conditions that could result in erroneous summarizations. However, the issues are dispersed throughout the literature in OLAP systems and statistical databases and have not been consolidated into a comprehensive taxonomy of issues that could result in inaccurate summaries. These analyses have focused on identifying the existence of erroneous summaries, rather than the impact on decision making. We note that the type

L. Delcambre et al. (Eds.): ER 2005, LNCS 3716, pp. 433 – 448, 2005.

of query and proximity of the summary outputs to decision points also affect the impact of inaccurate summaries. Furthermore, not all issues can be eliminated and methods for eliminating the problems can restrict number and types of queries that can be entered. It is therefore important to identify the factors that could lead to inaccurate summaries and approaches to manage them.

In [4], we analyze various causes of non- and semi- additive data in OLAP systems, and suggest rules for identifying and managing these data. This paper expands our earlier work on additivity and look beyond simply identifying the existence of aggregate summary problems.

In this paper, we present a taxonomy of inaccurate summary factors and practical rules for handling them. The primary contributions of this paper are as follows: first, our taxonomy of inaccurate summaries is comprehensive in that (1) we cover them from the perspectives of schema, data, and computation, and (2) we consolidate relevant terms and concepts in statistical databases with those in OLAP systems. Second, we suggest metadata that can be used to identify schema, data, and computational problems and suggest how to use this metadata to detect the impact that an invalid summary may have on a decision. Third, we present practical rules that can be used to quickly identify problems that have the potential to impact decisions:

Our paper is organized as follows: Section 2 describes relevant literature. Section 3 details a comprehensive taxonomy of summarizability issues in OLAP systems. Section 4 examines how these issues influence decision-making. Section 5 suggests techniques for managing summarization problems. Finally, Section 6 concludes our paper.

2 Background and Related Literature

Data warehouses are typically conceptualized as facts and dimensions, whereby facts are measures of interest, and dimensions are attributes used to browse, select, group, and aggregate measures of fact tables. Attributes that are used to aggregate measures are labeled classification attributes, and are typically conceptualized as hierarchies. An example of a classification hierarchy is the time dimension, upon which measures can be aggregated from the lowest level of granularity, dates, into progressive higher months, quarters, and years. For example, a profit measure may be aggregated from the daily profit to the monthly, quarterly, or yearly profit.

Typically, data is aggregated along multiple hierarchies, summarizing data along multiple dimensions. For example, a summary may show the total sales in the year 2004 at all branch locations in Pennsylvania. In this case, the sales measure is rolled up along the time dimension and location dimension. Because of the enormous size of the data sources, operations are performed to summarize measures in a meaningful way. The typical OLAP operations include Roll-up, Drill-down, Slice, Dice, Pivoting, and Merging.

Data are most commonly aggregated using the SUM operator in OLAP systems [5]. Measures can be classified based on whether they can be meaningfully added along hierarchies in various dimensions. Specifically, measures are classified as non-additive, semi-additive, or fully-additive, whereby a measure is:

- **fully-additive** if it is additive across all dimensions;
- **semi-additive** if it is only additive across *certain* dimensions; and,
- **non-additive** if it is not additive across any dimension.

Previous research on summarizability has focused on three primary areas. The first is identifying problems that could lead to summarizability problems [2, 3, 4]. The second focus is on defining methods for eliminating issues that could result in inaccurate summaries [6, 7]. The final focus is on making these problems visible through conceptual models [7, 8, 12].

Classifying measures based on the number of dimensions along which they can be aggregated is useful for making visible where inaccuracies may occur. However, this classification scheme does not give insight into the reason why measures are not additive, nor does it focus on problems that could result using other aggregate operators. In our previous work [4], we analyzed the reasons why certain attributes were not additive along certain dimensions, and distinguished between temporally and categorically semi-additive measures.

Lenz and Shoshani [2] take a broader look at the problem of erroneous summaries, and identify issues that are applicable to various aggregate operations. They describe three necessary conditions for summarizability, including disjointness, completeness, and type compatibility. Lehner, Albrecht, and Wedekind [6] suggest normal forms for multi-dimensional databases that can be used to guarantee summarizability. The focus of their research is to ensure that a broad range of schemas can be designed to meet both the completeness and disjointedness summarizability conditions specified by Lenz and Shoshani [2]. Hüsemann, Lechtenbörger, and Vossen [7] also suggest an approach for designing data warehouses that avoids aggregation anomalies. The approaches of both Lehner et al. and Hüsemann et al. focus on eliminating all possible aggregation anomalies through normal forms.

Tryfona et al. [8] suggest incorporating summary properties into the conceptual modeling of data warehouses. Specifically, they note that measures should be classified as *stock*, *flow*, or *value per unit* because different properties behave differently with different summary functions. Additionally, they note that object properties, including strictness and completeness should also be modeled. By incorporating this information into the model, potential problems can be made apparent at the conceptual level.

Shoshani [3] notes that OLAP systems and statistical databases are quite similar, and compares the work done in both areas. He argues that data warehouses should include a statistical object data type. Furthermore, he states that this statistical object data type should support the semantics, operations, and physical structure of the multi-dimensional space, and must also manage metadata of the category values and hierarchical associations.

In our paper, we expand upon this research by creating a comprehensive taxonomy of issues that could result in summarizability violations.

3 Taxonomy of Inaccurate Summaries

In OLAP systems, inaccurate summaries have typically been categorized based on both the number of hierarchies that measures could be aggregated, as is the case with labeling measures as additive, semi-additive, non-additive. In the statistical database (SDB) community, the terms *Flow*, *Stock*, and *Value per Unit* are used to classify summarizability problems. There are similarities among the problems identified in both the SDB and OLAP communities. We clarify the relationships between the terms used to identify problems in OLAP systems with those in statistical databases in Table 1.

Table 1. Comparison of summarizability terms used in OLAP and Statistical databases

SDB Label	OLAP Label	Description	Examples
Flow	Type of Fully Additive Snapshot	Total of transactions that occurred during the time between snapshots	*Change* in account balance during period *Change* in population during period
Stock	*Temporally* Non-Additive Snapshot	The level recorded at the end of a snapshot period	Account balance at specific point in time Population snapshot at specific point in time
Value per Unit	Type of Non-Additive Fraction	A fraction or rate used in expressions to convert units.	Exchange rates, which are used to convert different units of financial currency Cancer rate per million people, which can be used to convert population to number of cancer cases

In addition to the problems described above, the utility of a summary is also partially dependant on the extent of missing, biased, and inaccurate data, the type and purpose of queries, the visibility of any problems, and the content of the data. Therefore, we present a comprehensive taxonomy that differentiates structural issues, data issues, and computation issues. This taxonomy is intended to illuminate properties of the schema, data, and aggregate operator that could lead to erroneous summaries.

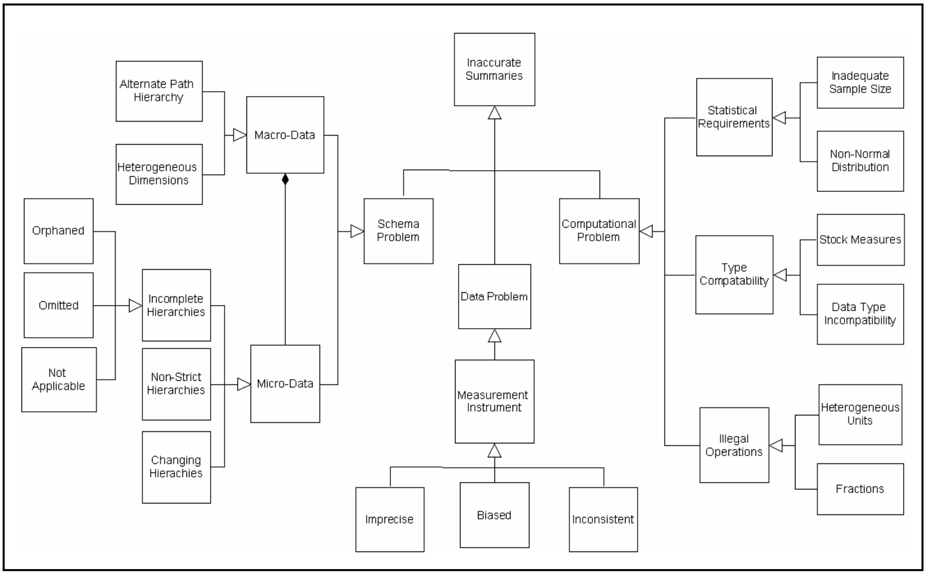

Fig. 1. The Taxonomy of inaccurate summaries in a UML class diagram

At the highest level, we differentiate three primary causes of aggregation problems: schema, data, and computation. *Inaccuracies due to schema* refer to those problems that are associated with the dimensional hierarchy, including non-strict and incomplete hierarchies, multiple path hierarchies, and heterogeneous dimensions. *Data-based inaccuracies* refer to problems related with the specific data instances, including changing data, inaccurate data, and imprecise or changing measurement

Table 2. Taxonomy of Potential Summarization Problems

Level 1	Level 2	Level 3	Description
Schema	Micro-Level	Non-strict	Hierarchy member has more than one parent
		Incomplete	*Orphaned*: Lower level hierarchy members do not have a parent in a higher level of a hierarchy
			Omitted: Null because real world data are not captured by the system
			Not Applicable: Null because there is no applicable value
		Changed	Hierarchy member splits into two members, merges into one, or moves from one parent to another.
	Macro-Level	Multiple Path Hierarchies	Lower objects in hierarchy have more than one aggregation path at higher levels
		Heterogeneous Dimensions	In schemas with multiple fact tables, the dimensions are not shared or exactly the same, but share a semantic relationship
Data	Measurement Instrument	Imprecise	Measurement instruments have inherent error associated with them
		Biased	Measurement instruments may capture data that are persistently higher or lower than the actual values
		Inconsistent	Aggregating data captured using different measurement instruments can result in erroneous summaries
Computation	Illegal Operations	Units (also referred to as categorically non-additive)	Cannot meaningfully aggregate data that have different units
		Fractions	Cannot meaningfully average measures derived from fractions, rather the numerator and denominator must be aggregated separately.
	Type Compatibility	Stock (Also referred to as temporally non-additive)	Stock levels (snapshot levels) are stored instead of flow (change over time) data; includes measures that represent averages, maximums, and minimums
		Data Type	Cannot use certain aggregate operators with some data types
	Statistical Requirements	Sample Size	Some operations require minimal sample sizes to be significant
		Distribution	Certain operators are more appropriate for normally distributed data, while other operators are more appropriate for non-normally distributed

instruments. And, *computational inaccuracies* refer to problems related to inappropriately computing aggregates summaries, such as those that could result from summing measures of intensity, summing data snapshots, or using the mean to find the central tendency of log-normally distributed data. Figure 1 summarizes the taxonomy in a UML class diagram and Table 2 depicts our taxonomy of potential summarization problems.

3.1 Schema Level

Inaccurate summaries can result when the structure of the classification hierarchies does not meet certain necessary conditions. Specifically, problems can result from non-strict and incomplete hierarchies, and can occur at the level of either micro-data or macro-data. Micro-data refers to base data, while macro-data refers to schema objects.

A strict hierarchy refers to a classification hierarchy whereby each object at a lower level belongs to only one value at a higher level. *Non-strict hierarchies* can be thought of as many-to-many relationships between a higher level of a hierarchy and a lower level. Lenz and Shoshani [2] refer to strict hierarchies as disjointed, and note that disjointedness of category-attributes is a necessary condition for summarizability. In order to test for disjointedness, or hierarchy strictness, it is necessary to examine the semantic knowledge of the micro-data or test the actual data. They describe students being assigned to a single department as an example of disjointedness; whereas, if students could be assigned to multiple departments, the disjointedness property would be violated. The non-strict hierarchy in Figure 2 depicts a situation where at the schema level, the student object rolls up only to one higher level object, Department, but at the micro-data level, an individual student could be assigned to more than one department. If each student has values for *tuition_paid* associated with them, then the payments associated with these students may be counted more than once.

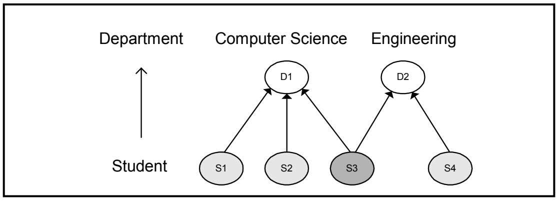

Fig. 2. Non-strict Hierarchy

In order to ensure that dimensional hierarchies are complete, it is necessary that they satisfy two conditions. All lower level members must belong to one higher level object, and that object must consist of those members only. We will refer to these different types of incomplete hierarchies as *Orphaned-incomplete*, which will include hierarchies where lower level records are stored, but are not associated with parents. *Omitted-incomplete* include hierarchies where records are not stored in the database.

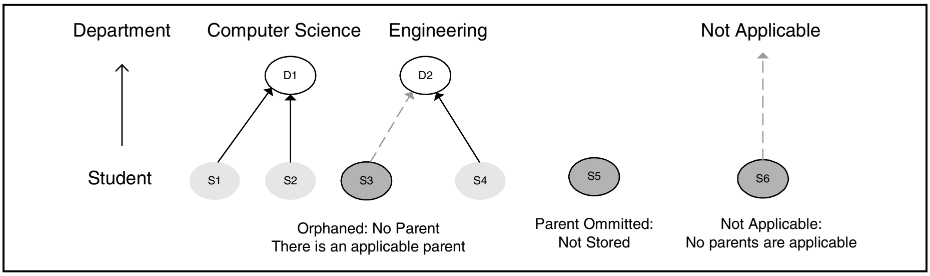

Fig. 3. Three Cases of Incomplete Hierarchy

Additionally, we will also differentiate null values that occur when there are no applicable parents. We label these missing values as *Not Applicable-incomplete*. In the student-department example, there are three situations where the hierarchy does not meet the completeness property. First, it is possible that a student was stored in the database, but their department was not stored in the database. Second, it is possible that the student existed and was assigned a department, but was not stored in the database. Finally, it is possible that the student is not and should not be assigned a department, as may be the case with non-matriculated students. In any case, aggregating the data along this hierarchy will be incomplete. Figure 3 depicts an incomplete hierarchy of the three cases.

Thus far, we have looked at situations where one dimensional category is rolled up to a single dimensional category (i.e. student rolled up to department). *Multiple path hierarchies* consist of hierarchies where a lower level category can be aggregated to more than one higher level category. In situations lower level instances are associated with more than one parent which are located in different dimensional attributes. In some cases, instances can only have one parent; while in other cases, instances can have parents in multiple different attributes. For example, Hurtado and Mendelzon [9] describe an example data warehouse schema, whereby an international organization keeps track of shops, which have a parent in either state or province, depending on their country. It is legal to aggregate the sales in a select set of states with those in a select set of provinces. However, if a child has parents in both of the selected attributes, then erroneous summaries can result.

This situation could be further complicated if these parents are part of different heterogeneous dimensions. This situation could happen in multiple data marts. Data marts can be tied together using drill-across techniques when dimensions linked to the facts are either exactly the same or perfect subsets of each other. Abelló, Samos, and Saltor [10] argue that completely conforming dimensions unnecessarily restricts the usage of drill across. They argue that it is possible to drill-across different fact tables if there are derivation, generalization, association, or flow relationships among the related dimensions. While it may be possible to relate fact tables that do not share dimensions, drilling across non-conformed dimensions could result in inaccurate summaries. Abelló et al. [10] note that dimensions evolve over time, and new schemas can be linked to older schemas through flow relationships. However, semantic relationships may differ at different times. For example, in [4], we describe how the dependency between area code and location was eliminated, making it possible to misinterpret comparison queries between facts stored before and after the dependency changed.

3.2 Data Level

Imprecise, biased, and inconsistent data may result in erroneous summaries. This is especially true when some measures stored in data warehouses are derived from *measurement instruments*. Measurement instrument is a term used to describe the method used to collect the base data. Measurement instruments can be physical instruments (e.g. thermometers, photometers, GPS systems, registers, bar-code scanners), or they can be methods for collecting data (e.g. census surveys, inventory counts).

Regardless of the medium used to collect the data, all measurement instruments have some imprecision associated with them. Imprecision refers to the exactness or reliability of the data collected from a measurement instrument. Bias is a measure of the systematic offset or shift of data collected from a measurement instrument. If the values captured are persistently lower than the data in the real world, then the measurement instrument is considered to be negatively biased. Alternatively, if the captured data are persistently higher than the real world value, the measurement instrument is positively biased.

Inconsistency occurs when the method used to record data changes because the physical instruments used to record the data were changed, the software used to capture the data changed, or the procedure used to record or capture the data was altered. Aggregating data that was collected using different measurement instruments (including process changes) can result in erroneous summaries. And, data collected with one measurement instrument may not be comparable with data collected using a different measurement instrument. Unless changes are recorded, decision makers may come to the conclusion that there were changes in the data; when, in fact, the differences were only a result of the measurement instrument.

3.3 Computation Level

Computational inaccuracies are problems that are related to using aggregate operators that are not appropriate for the data, or aggregating data with differing units. Rafanelli, Bezencheck, and Tinini [11] classify three types of meta-data that are considered relevant to summarizability, including the aggregation function (count, sum average etc.), the summary type (real, non-negative integer, etc.), and the phenomenon described by the statistical object (population, income, etc.). Computational problems occur when there is an illegal or problematic interaction between these different components. Specifically, an issue arises when summary objects cannot provide meaningful summaries of the phenomenon described by the statistical object using certain aggregate operators. Several types of computational problems are described below.

Lenz and Shoshani [2] distinguish between stock and flow measurements. Measures that can be characterized as stock measurements, such as inventory levels or bank account balances, are typically non-additive across the time dimension, meaning that these measures cannot be aggregated using the summation operator unless grouped by time. We named these measures temporarily non-additive [4]. Certain stock levels such as measurements of intensity, and pre-aggregated averages, maximums, and minimums, cannot be meaningfully added regardless of the elements by which these measures are grouped [5], [4]. While stock levels cannot be aggregated using the sum operator, all other aggregate operators can be used to aggregate stock measures. Other measures cannot meaningfully be aggregated using either the sum or average operator, such as measures of direction [4].

Certain measures cannot be meaningfully aggregated regardless of the aggregate operator. It is not mathematically permissible to aggregate values that have different units. In a scientific context, an example would be attempting to aggregate a measure of distance with a measure of mass. And, it is not permissible to aggregate two different measures of distance with different units unless the data are translated into a common unit. In a data warehousing context, basket counts cannot always be meaningfully aggregated because these measures aggregate items with different units. We named these measures categorically non-additive [4]. For example, a basket-count may aggregate 1 telephone with 3 packs of bubble-gum for a total of 4 items. In this case, the data are translated into similar units based on an abstraction hierarchy, whereby both bubble-gum and telephones are products. As we discuss in [4], certain basket-counts may be meaningful if there is a low level of abstraction necessary to translate the items into similar units.

In [5], Kimball and Ross note several types of measures that are inherently non-additive. They state that percentages and ratios, such as gross margin, are non-additive, and therefore, when designing systems, both the numerator and denominator should be stored in the fact table. Additionally, they note that it is important to remember when summing a ratio, it is necessary to take the ratio of the sums and not the sums of the ratios. In other words, the numerator and denominators should first be aggregated separately using the sum operator, and then the totals should be divided, yielding the appropriate ratio values.

4 Measuring the Influence of Inaccurate Summaries

Incorrect summaries can result if instances of measures are incorrect, not counted, or counted more than one time in an aggregate summary. It is not always possible to automatically identify and record sources of summarization problems, since many problems are derived from imprecise business process, measurement instruments, or human errors. Therefore, it is important to identify the *influence* that the erroneous results that may not be eliminated during the design process will have on decision making.

We note that not every summarizability violations may lead to inaccurate decisions. Some problems are insignificant, and would have no effect on the decision process; while others are quite significant and could result in severe mistakes. Eliminating all possible inaccurate results can be overly restrictive and effortful; alternatively, ignoring the potential for inaccuracies could lead to serious errors. One approach without making the system less restrictive is to identify summarization problems in conceptual models [12], [8]. While modeling issues is useful for depicting the existence of a problem, it does not show the influence that an issue may have on the decision-making process. The influence of inaccurate summaries on decisions can be classified along two dimensions, extent and impact. Whereby,

- *the extent* refers to how many decisions will be affected by a particular issue; and,
- *the impact* refers to the degree to which an issue will affect a particular decision.

The extent of inaccurate summaries refers to the scope of decisions that are affected by an inaccurate summary. It is not possible to precisely predict the extent of a summarizability issue because the exact queries that will be run against a schema cannot be precisely known ahead of time. The impact of inaccurate summaries is even more difficult to identify or estimate due to many important factors. In this paper we briefly discuss a way of estimating the extent of inaccurate summaries.

The *extent* of summarizability problems can be measured performing an analysis on the types of queries affected. To ensure a comprehensive analysis, each measure should be analyzed along each hierarchy using all aggregate functions. And, potential sources of biased, missing, or duplicate data, such as those described in Section 3, should be noted. Following this process of identifying all possible sources of erroneous data will be time consuming, but there are several ways to facilitate the process. One suggestion is to automate the process of identifying problems. Grumbach and Tininini [13] suggest a method of tracking numerical dependencies and using metadata to automatically aggregate data that are not necessarily complete. Hurtado and Mendelzon's [9] method of using summarization constraints using metadata can also facilitate the process of making the extent of inaccurate summaries visible.

Queries on data warehouses can be classified according to the decisions they are intended to support. Specifically, queries can be categorized into exploratory queries, comparison queries, and benchmark queries. *Exploratory queries* are used to get a general idea of the data. *Comparison queries* are used to compare distinct sets of data from the system, comparing summaries from different groups. And *Benchmark queries* are used to compare sets of data against a specified benchmark. The type of queries affects both the impact that an erroneous summary will have on a decision and the approach for managing the problem. Examples of benchmark queries, comparison queries, and exploratory queries are shown in Figure 4, 5, and 6, respectively.

How many of our branch stores met the profit goal of $5,000,000 last year?

Which regions of the country have cancer rates that are greater than one in a million?

Which programs have not shown a profit in two out of the previous five years?

Fig. 4. Benchmark Queries

Are electronics sales more profitable than appliance sales?

Which region of the country has had the most new customers during the previous six months?

Are there more cases of influenza this year than the average number of cases during the previous 10 years?

Fig. 5. Comparison Queries

How are the sales in North America doing?

How many people have contracted the HIV virus during the past year?

Fig. 6. Exploratory Queries

With comparison analysis, the impact of erroneous summaries will be most apparent if the bias of one group of data is different from the bias of another group. For example, if both appliance sales and electronics sales are positively biased by the same amount, comparisons between the two will not affect the decision, even though the data are inaccurate. However, if the total sales displayed for appliance sales is positively biased and the total sales for electronics is not, then an inaccurate decision could result if the bias is significant enough to change which group was more profitable.

In both benchmark and comparison analyses, erroneous decisions may be made if the error associated with the aggregate summary cause the displayed value to be on a different side of the decision cut-point than the true value. In exploratory analyses, there may not be any definable decision points at all.

In a sense, OLAP queries can be conceptualized as a type of informal statistical test. Statistical tests are used to determine whether groups are different from each other or a benchmark, and use the error to identify whether the output is reliable at a certain significance level. When running statistical tests, rigorous rules for identifying significance are needed. These rules are based on the probability of an output being incorrect based on the characteristics of the dataset. In data warehousing, it is also important to identify the reliability of an output, and similar considerations should be given to identifying whether the error renders an output insignificant for a given test.

5 Managing Inaccurate Summaries

In this section we suggest an approach that can be used to minimize and identify the impact that inaccurate aggregate summaries will have on a given query. We also suggest techniques and metadata that can be used to automatically detect and display their effect on analyses.

5.1 Schema Level

Much of the work on summarizability has focused on either structuring data warehouses in a manner that eliminates the potential for structural violations to occur or making the violations apparent through conceptual modeling. These techniques are useful for eliminating many inaccurate summaries from occurring or impacting decisions. However, there are many situations where systems are not optimally structured or conceptual models are not consulted. In these situations, it is important to make the impact of structural violations apparent at query time.

Scripts can be run that count the number of times a value is included in a summary. This value can be used to identify orphaned incomplete data or duplicate data resulting from non-strict and alternate path hierarchies. Eder, Koncilla, and Mitsche [14]

Table 3. Managing Schema Problems

Type	Queries Impacted	Management
Non-strict	Any query that rolls-up to the parent level of a non-strict hierarchy will be affected only if the lower level values are counted more than one time in the query	Run scripts at query time to identify the number times measures are counted in the summary and identify the total value of any duplicated values.
Incomplete: Orphaned	Any query that aggregates parents of orphaned data will be impacted only if the orphaned values are associated with measures.	Routinely run scripts to identify the extent of orphaned data and the value of the associated measures.
Incomplete: Not Applicable	Any query where it is important to distinguish whether a value is missing, orphaned, or not applicable	Use a not-applicable code when there is not an appropriate dimension member, rather than leaving the cell null.
Incomplete: Missing	Any query with missing data can significantly impact queries, especially when data are systematically missing	Attempt to identify reasons for missing data and extrapolate the impact of the data that were not captured.
Changing Schema Dependencies	Comparison queries that compare temporally dissimilar groups of data	Identify and track all hierarchical dependency changes in metadata.
Alternate Path Hierarchies	Merge queries will be inaccurate if data are counted multiple times. Comparison queries may count a single measure in more than one group.	Run scripts at query time to identify whether measures are counted in multiple groups or multiple times in a single group.
Heterogeneous Dimensions	Drill-across queries where the dimensions are not perfectly conformed	Dimensions should be conformed if significant inaccurate summaries result from heterogeneous dimensions.

describe the applicability of using regression, correlation, Fourier transforms, and principal component analysis to identify sharp changes in the structure of data warehouses. Specifically, they explore the use of these outlier detection algorithms for identifying hierarchical members that split, merge, change, or have moved. When inaccurate summaries result from heterogeneous dimensions, we recommend that Kimball and Ross' [5] suggestion of the conforming the dimensions be followed. Table 3 shows our suggestions for managing schema problems.

5.2 Data Level

Inaccurate summaries could significantly affect decisions in both business applications and in scientific database applications. Typical sources of errors include units, capture biases, errors due to capturing frequencies of stream data, etc. When the method used to capture and record measures changes during the history of data collection, the resultant data can be affected. Therefore, it is important to make heterogeneous measurement instruments visible to decision-makers. To reduce the likelihood of

Table 4. Managing Data Problems

Type	Queries Impacted	Management
Biased	Any exploratory query that aggregates biased measures Comparison queries where the groups are biased in different directions	Track the positive or negative bias associated with each measure, method, or measurement instrument.
Imprecise	Any query where the aggregate value is close to a decision point	Display the level of precision associated with a summary. Indicate the likelihood that an aggregate value is significantly above or below that value.
Inconsistent	Comparison queries that aggregate measures captured using dissimilar methods or measurement instruments	Store method code indicating to track how a particular measure was captured. Also, track how the dimensional members were captured.

erroneous conclusions based on these summaries, each different measurement instrument should be stored in metadata along with the associated measures. When aggregate queries are run, these metadata should be accessed. And, if there are multiple measurements within a single sub-cube or among different sub-cubes, then there is the potential that the measurement instrument affected the aggregate results.

When biases are known, the strength, direction, and extent of the bias should be stored in metadata. These metadata should then be accessed to adjust data based on these biases. There are several likely sources of bias, including data being duplicated, missing, or inaccurate. Queries that pull biased data should automatically adjust the data to eliminate the bias. We distinguish between three types of biases: missing, duplicate, or shifted.

Often, biases in a data set, however, cannot be precisely known. In these cases, it is also important to track the precision associated with the measurements. These precision values can be used to display a summary output that displays a range of values that could be representative of the true value, rather than a single imprecise value.

The impact of the imprecision and bias must be identified if decisions are going to be based off of these results. The total bias and imprecision should be combined to determine the offset and error associated with a summary. This information can then be used to measure the impact that these issues will have on a decision. Data problems can occur from imprecise, biased, or inconsistent data. To manage these problems, we suggest storing the error and offset for values associated with each measurement instrument. The method used to capture the data should be stored and used to distinguish among values captured using different processes or systems. Table 4 summarizes our suggestions for managing data issues.

5.3 Computation Level

Computational problems may impact decisions, especially when the violations are not apparent to the decision maker. Aggregating basket counts may result in an erroneous summary if dissimilar items are counted together. Averages performed on non-normally distributed data may give an incorrect perception of the central tendency.

And when specific rules are prescribed for aggregating data, erroneous decision may be made if these rules are not apparent to the decision-maker.

All measures have associated units, such as dollars, degrees, inches, or product types. The units must always be stored in metadata. It is best to store the units as a dimensional field or in a conformed dimension called Unit in the data warehouse. The Unit dimension should have also necessary conversion rules. Aggregate operations cannot be performed on measures with differing units unless they can be converted into a common unit. It is also important to check inter-set heterogeneity in comparison queries. The greatest impact will occur when the heterogeneity of the units is not apparent to persons performing the query. This type of error may occur if a single measure stores currency with multiple financial units that have near one-to-one exchange rates.

If queries are found to have data with more than one different unit, they must be transformed into similar measures. In cases where units can be assimilated computationally, this can be done automatically by storing conversion units in meta-data or in Unit dimension. Analysts will be able to aggregate the data by simply choosing the units for the data.

Many other computational problems result from performing illegal operations, and therefore should not be permitted. Systems should track whether fields are fractions, measures of direction, and stock values; and, queries that attempt to improperly aggregate these data should be prohibited. Table 5 depicts our specific suggestions for managing computation problems.

Table 5. Managing Computation Problem

Type	Queries Impacted	Management
Illegal Operations	Queries aggregating measures with heterogeneous units that appear to have similar units are most likely to be misinterpreted	Units for all measures should be stored in metadata and in a conformed dimension; scripts should be run at ETL stage or query time to convert units if heterogeneous units exist. Queries should be grouped by the units so that no summaries will aggregate measures with different units.
	Queries aggregating measures that are derived from fractions, such as Gross Margin Return on Investment (GMROI)	When using the sum operator, fractions should be aggregating by taking the quotient of the sums, rather than the sums of the quotients.
Type Compatibility	Sum Queries that aggregate snapshots of stock measures. Any query that aggregates data using an inappropriate aggregate operator, such as measures of direction.	Appropriate aggregate operators for each measure should be stored in metadata.
Statistical Requirements	Aggregations that are used for statistical calculations that have specific requirements	Show alerts when aggregate summaries are based on very limited number of instances. Analysis tools should show distribution of data and descriptive statistics for the summaries.

6 Conclusions

In this paper, we presented a taxonomy of inaccurate summary factors and practical rules for handling them. We discussed these issues from the perspectives of schema, data, and computation. We proposed several methods that can be used to identify problems based on the type of queries that will be run. Finally, we suggested metadata and practical rules that can be used to manage inaccurate summaries.

We note that not all summarization problems can be eliminated from OLAP systems. Furthermore, methods for eliminating and managing summarization problems can be effortful. Therefore, it is important to prioritize problems based on how likely they are to impact decisions.

The following heuristic rules can be used to quickly identify problems that have the potential to impact decisions:

- *Follow design guidelines wherever possible*
  Conforming dimensions and making dimensions orthogonal is an important step to reducing the likelihood for misinterpretation of aggregate summaries.
- *Identify inconsistencies*
  Beware of aggregating data that has been collected through different methods or collection instruments. It is also important to identify measures that are collected using the same method that are coded differently.
- *Link all measures to their corresponding units*
  The units of all measures should be stored either in the data warehouse or in metadata. Additionally, if the units associated with measures are part of a hierarchy, the associated hierarchy should also be stored. Measures should only be aggregated with measures that have similar units.
- *Make imprecision and bias visible*
  Where it is not advantageous to completely eliminate the potential for inaccurate summaries, systems should allow decision-makers to make more informed decisions by making error and offset values accessible through links to visual or tabular outputs.
- *Track the dimensions along which measures are non-additive and non-summarizable*
  OLAP systems should then display alerts or prevent queries that attempt to improperly summarize these measures.
- *Make computational assumptions and operations visible*
  OLAP tools often hide the computational aspects of aggregating data, such as rounding rules, equations, and statistical assumptions. This information should be directly accessible through OLAP tools so analysts can quickly and easily identify potential computational problems.

This paper contributes to a comprehensive understanding of summarizability and their impact on decision-making. Identifying source of errors and learning how to manage them can reduce unnecessary effort of imposing overly restrictive rules to eliminate all summarizability violations. Our paper gives designers a means to manage and prioritize inaccurate summary problems.

References

1. Martyn, T.: Reconsidering Multi-Dimensional Schemas. *SIGMOD Record*. Vol. 33, No. 1. ACM Press. New York, NY (2004) 83 – 88.
2. Lenz, H-J. and Shoshani, I.: Summarizability in OLAP and Statistical Data Bases. *Ninth Int'l Conf. on Scientific and Statistical Database Management* (1997) 132-143.
3. Shoshani, A.: OLAP and Statistical Databases: Similarities and Differences. *Proc. of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems,* Tucson, Arizona (1997) 185 – 196.
4. Horner, J., Song, I.-Y., and Chen, P.: An Analysis of Additivity in OLAP Systems. *DOLAP'04*, Washington, DC, USA (2004) 83-91.
5. Kimball, R. and Ross, M.: *The Data Warehouse Toolkit*: Second Edition. John Wiley and Sons, Inc. (2002).
6. Lehner, W., Albrecht, J. and Wedekind, H.: Normal Forms for Multidimensional Databases. *Proc. of the 10th International Conference on Scientific and Statistical Data Management (SSDBM'98)*, Capri, Italy (1998) 63-72.
7. Hüsemann, B., Lechtenbörger, J., and Vossen, G.: Conceptual Data Warehouse Design. *Proc. of International Workshop on Design and Management of Data Warehouses* (2000) 6
8. Tryfona, N., Busborg, F., and Borch Christiansen, J.: StarER: A Conceptual Model for Data Warehouse Design. *DOLAP '99*, Kansas City, MO, USA (1999) 3-8.
9. Hurtado, C and Mendelzon, A.: OLAP Dimension Constraints" *ACM PODS 2002*, Madison, WI, USA (2002) 169-179.
10. Abelló, A. Samos, J., and Saltor, F.: On Relationships Offering New Drill-across Possibilities. *DOLAP '02*. McLean, VA, USA (2002) 7-13.
11. Rafanelli, M., Bezencheck, A., and Tininini, L.: The Aggregate Data Problem: A System for their Definition and Management. *SIGMOD Record*. Vol. 25, No. 4 (1996) 8-13.
12. Trujillo, J., Palomar, M. Gomez, J., and Song, I.-Y.: Designing Data Warehouses with OO Conceptual Models. *IEEE Computer*. Vol. 34, No 12. (2001) 66-75.
13. Grumbach, S., Tininini, L.: On the Content of Materialized Aggregate Views. *ACM PODS*, Dallas, TX, USA (2000) 47-57.
14. Eder, J., Koncilia, C. Mitsche. D.: Automatic Detection of Structural Changes in Data Warehouses. *DaWaK* (2003) 119-128.