December 2006

Data Quality: An Attribute Framework for Large Systems

Ann Robbert
Bentley College

Linda Senne
Bentley College

Data Quality: An Attribute Framework for Large Systems

Mary Ann Robbert
Bentley College
mrobbert@bentley.edu

Linda Senne
Bentley College
lsenne@bentley.edu

ABSTRACT

In previous work we examined a process for cleaning data from a single source. We proposed a framework with three attribute categories—critical, logical, irrelevant—that enables users to interpret the effect of dirty data on operational decisions. To extend the framework to large systems, we suggest adding a fourth category "used" as a set of tags. The metadata on the tags describes the usage—how users modified the attributes to meet their requirements.

One database we examined contains data feeds multiple sources within an international firm plus external feeds containing financial market information. As the data flows through the corporation to the users, additional attributes are added and multiple databases are integrated. At each stage users endeavor to maintain clean data for attributes important to their function. Downstream analysts, however, do not know which attributes have been used. This paper describes the framework extension and the metadata tags.

Keywords

Data quality, quality framework, large systems, usage

INTRODUCTION

Information quality (IQ) is defined as information that is fit for use by information consumers (Strong, Lee, Wang 1997). However, users in an organization define IQ differently depending on the needs of their function. As a result, staff in one department may consider data "good" while another department may not because each department can use the same records in the database for entirely different purposes. We are naming this usage.

Users may not have control over IQ because they do not control the data entry process. In one of the large organizations we studied, hundreds of lines fed data into the system. This data was typically cleaned when it entered the system or it came from a trusted source. However, once the data was in the system, it flowed into and through many subsystems where it was used "as is" or integrated with other data sets. Users at various stages of the pipeline could not be sure what processes had been used to clean the data initially nor could they be certain of the data source. Furthermore, they could not determine whether the data had been modified as it flowed from one department to another. In fact, if changes had been made, they frequently were unable to ascertain how or why these changes had been implemented.

In a prior paper we developed a framework to help users analyze the significance of the attributes in a database (Senne, Robbert,Haughton 2005). The framework classifies the attributes in a set of records into three categories:

- Critical: attributes that must be used to make a given business decision; if these attributes are inaccurate or missing, the quality of the decision suffers
- Logical: one of several attributes that users may choose to help in the decision process if they can verify the quality of the data
- Irrelevant: not a significant factor.

This framework works well within a single group of users who can focus on cleaning the data to their specifications. Users can employ an iterative process to work on the data until it is sufficiently clean. However, in a large organization where databases are part of a pipeline of information flow, iterative techniques for cleaning the data at the attribute level are not adequate. This paper expands the framework to include a means to identify the attributes that have been modified or never used. We define a tag-table that quickly reveals the history of an attribute as it flows through the system and alerts users to metadata information that they should monitor. The tag-table provides users throughout the data flow process a means to easily identify the current state of attributes and whether the attribute has been used. This paper focuses on information that is stored in the tag-table and in the metadata file that assists users in identifying data quality problems.

RELATED RESEARCH

Our premises are similar to those of Lee who defines "know-what" knowledge as the information necessary for achieving high intrinsic information quality (Lee, 1996). "Know what" knowledge focuses on goals and resultant activities of information quality to improve accuracy and believability in the information provided to users. Our framework relies on knowledge of attributes use. Morey points out that the need for accuracy differs drastically with different types of records (Morey, 1982). In addition to record type, we feel the need for accuracy also depends on usage. Orr supports this point; he lists as a data quality rule: "Data quality in an information system is a function of its use, not its collection" (Orr 1998).

The metric for missing data cannot be just evaluated on number of times values are missing. Users need to know how the missing value affects business, usage, or even if any "notion of IQ depends on the actual use of data" (Wand, Wang 1996). The metadata contains information on how null values were derived by different users.

By extension, attributes that are not used cannot be assumed to be correct. Orr noted "Data that is not used cannot be correct for very long," (Orr 1998) and Vassiliadis et al notes that, when no one uses the data, no one will take care to improve its quality(Vassiliadis,Bouzeghoub,Quix 1999). Thus we tag any attribute that has not been used.

Our selection of metadata is supported by Rothenberg who suggests users augment databases with metadata in order to record information needed to assess the quality of their data (Rothenberg). He gives a detailed discussion of metadata in general, the levels of metadata and how metadata could be augmented to store IQ information. Vassiliadis examines the evolution of data and suggested complementary metadata to track the history of changes and provide a set of consistency rules to be enforced when reevaluating a quality factor (Vassiliadis et al 1999). For this paper we are using Hert's definition of metadata as information used to interpret, use and understand the information (Hert 2001).

FRAMEWORK DEFINITION

In a prior paper the authors defined a framework for determining the significance of attributes to a decision maker (Senne, Robbert,Haughton 2005). A logistic regression model was used to prove the benefits of using the framework on historical mortgage data.

In the framework, the attributes—critical, logical, irrelevant—affect how a user determines which attributes to use in a business decision and whether their values need to be cleaned. It is important to focus initially on just critical attributes—those attribute that, if they are missing or inaccurate, lead to a poor decision. The figure indicates appropriate actions for users during the cleaning process.

Number of Records / **Attribute**	**Small Number of Records with Null, Incorrect, Inconsistent or Counterintuitive Values**	**Large Number of Records with Null, Incorrect, Inconsistent or Counterintuitive Values**
Critical	Eliminate records	Data set too dirty for the basis of a decision
Logical	Eliminate records and/or exclude attribute from decision	Exclude attribute from decision

Figure 1. Framework for Handling Critical and Logical Attributes

Users within a single department can follow an iterative process to clean the data. First, users decide whether *records* need to be eliminated. If too many records must be dropped, the data may be so dirty that it cannot be used in operations. To determine if records should be eliminated because critical attribute values contain dirty data, users ask whether these critical values

a. Are *null* in some records?

b. Appear to contain *incorrect values*?

c. Seem *counterintuitive or internally inconsistent?*

If the answer to any of these questions is "yes," they should eliminate the record.

In the second step, users examine individual attributes in the remaining records. Users ask the same questions—are the values null, erroneous, counterintuitive or internally inconsistent—but in this pass they focus on attributes that, *based on prior knowledge, logically* might *help make a decision*.

a. Some records contain attributes with *null* values. For example, at a college, students denied admission may have null values for attributes that are entered only for students who have signed an acceptance form. Eliminating these records would eliminate valuable information about who applied. In this case, as long as the attributes are not critical, they should be excluded from the decision making process but all the records retained.

b. Logical attributes in some records appear to contain *incorrect values* such as an interest rate of 48% on a mortgage. Such records can be dropped from the data set but the attribute retained.

c. Some logical attributes seem *internally inconsistent* or *counterintuitive* such as a significant number of mortgages in the data set that are significantly higher than the value of the property. These records with inconsistent attributes should be eliminated from the data set only if the process does not leave the user with too little data for decision making. If too little data remains, the attribute cannot be used in decision making.

While this iterative process may be sufficient for some users, it is not adequate for those who receive data as part of a flow from multiple systems.

PROBLEMS INHERENT IN LARGE SYSTEMS

Many separate organizations, both internal and external, affect the complex systems in large organizations. We examined many systems including one with 1,500 input streams. Frequently, the input streams are integrated at many different points as the data flows from user group to user group.

Large systems, often located across the world in various time zones, contain subsystems that serve different goals and needs. The users of each subsystem clean or simply use available data to meet their requirements. During each operational phase

- Some or all of the attributes can be fully or partially cleaned
- Derived attributes can be calculated and inserted
- Mathematical calculations to estimate null values can be executed
- Data can be added or integrated. This data might be unused, cleaned or derived.

In most cases these activities are carried out independently within an organizational unit.

Groups using the subsystems are not necessarily co-located nor are they available to answer questions. Consequently, more information than just a description of the attributes and their values needs to accompany the data as it moves downstream. While users need to know whether data has been modified, it is equally important to know whether data—and individual attributes, in particular—has been used because data cleanliness is proportional to its usage.

LIMITATION OF FRAMEWORK

In the context of complex, dynamic systems we believe the framework should be expanded to follow the data flow, adding data quality information as part of the metadata to help the user understand the cleanliness of the data they are managing. This quality information can be easily tracked with a tagging system that can be queried to indicate the usage status of an attribute. All attributes are stored as records in a tag-table. The number of binary tags that can be used is flexible and not subject to size constraints.

Many groups in the organization use the same data and tags indicate various conditions of the attributes:

- evaluated as critical
- evaluated as logical
- used
- cleaned
- integrated or modified at some point or not, i.e. the attribute contents have been merged from multiple sources or changes have been made to the formula in the case of a derived attribute
- eliminated at some point.

Users can add other tags, e.g., values inserted for nulls, contains estimated values, as needed to meet specific requirements. Note that these tags are used for a quick overview of the data. Organizational ID's, dates, specific changes and other details are recorded in the metadata and can be accessed from there.

The simple tag-table contains at a minimum these six tags as columns and the names of all attributes from the tables serve as row identifiers as shown in Figure 2 for mortgage data. In the sample table the first two columns indicate whether the

attribute is critical, logical or if not checked, neither. This is a disjoint set. Usage is then indicated. If the attribute has not been used neither of the next two columns can be checked. Information about attributes that are eliminated by any set of users should be retained in the metadata. All attributes continue to be listed in the tag-table with the "Eliminated" box checked when an attribute is removed.

Attribute /Activity	Critical	Logical	Used	Cleaned	Integrated modified	Eliminated
ID	X		X			
Name		X				
Date		X	X			
Balance		X	X		X	
Monthly Payment	X		X	X		
Term	X		X			

Figure 2 Sample Tag-Table

The table name/attribute name pair is a unique identifier. All tag values are initially set to false. An application program or procedure changes the values to true when a specific event occurs.

If this technique were implemented at the data instance level, increased detail would be available to the user. However, the cost of implementation would likely be prohibitive since the rows would have to contain each value of an attribute, not just the name of the attribute itself. As a result, the number of rows would increase dramatically actually explode. The columns as proposed need to contain only one character tag, but if data level values were included we might want to include whether a specific value is as entered, has been changed, is null, or was derived, increasing the number of columns in the table by the number of possibilities.

Attributes with a true tag to indicate eliminated records need special treatment. The user has the option to either reinstate the records tagged as eliminated and located in the metadata or leave the records out of the data set. Thus the user in a particular subgroup has the ability to re-include the records if needed or to reconfigure the attributes.

In summary, the tags are flags to lead users to metadata where they can easily see the attribute usage status as well as easily find further information on the data set.

METADATA

A metadata repository can be used to provide the information required to understand quality goals and quality factors. Metadata tracking follows the flow of data through the system to the different user groups. The purpose of metadata is to supply context for data. It contains the original context and documentation for data as well as a description of the process, assumptions, and conditions for generating new data. Metadata can also be used for evaluating and recording data quality. More than just listing constraints and use, the metadata should include a discussion of each dataset, restrictions on use and access, processes that use the dataset, and rationale for decisions and assumptions including who made them when and why.

The initial completeness of the metadata and the dynamics of the data can be sources of major problems. Changes are not always added even if the metadata starts off complete and correct because those who must make the entries may, for example, be under time pressure or not understand the significance of their actions. Another problem mentioned by Dasu and Johnson is the independent existence of the data and metadata where the DBMS metadata facilities are not always used and a minimum number of data descriptions are documented (Dasu, Johnson 2003). The model and its upgrades as well as the data must be included in the metadata. Data transmitted from group to group in delimited flat files must be documented upon addition to the database.

Since metadata is a permanent record, no re-entry is necessary. However, as the data flows to different groups, metadata must be available throughout the system in the same format. Since users do not control the metadata, the information capture must be automated.

Metadata items can be descriptive, supplying an objective attribute of a data item, or evaluative, providing an assessment of how appropriate a data item is for some purpose. Metadata communicates contextual information; ideally it contains "enough context to allow a subject-matter expert to evaluate its appropriateness for a specific purpose without additionally requiring that expert or user to be knowledgeable about the idiosyncrasies of the database"(Rothenberg).

The rationale for eliminating records or changing values is also contained in the metadata. For example, consistency checks are defined within the metadata. Value distributions that are known for a given data set are also documented. Relationships between one data item and another data item either in the same or other databases are also recorded. Summary information should also be stored in the metadata.

There is a detailed linkage between objective quality characteristics and user dependent goals. For example, the metadata should describe the quality of particular data items as well as results of overall assessments performed on items. Assessment details need to be recorded for different data uses. The same data set might be assessed differently for different uses. The intended use must be described to determine the success of the evaluation. The quality of the database as a whole must be measured either with respect to the specific purpose of each organizational group and/or in general. Ideally the measure of data value, tracking replication, and utilization are also maintained in the metadata. Since quality can be assessed more than once, the assessment results need to be tracked over time and stored in the metadata. Metadata should likewise track the expected degradation of elements over time.

It is not only necessary to track changes to the structure, content or meaning of data, but we must do the same for the metadata. It should be noted that metadata will increase in the downstream direction. In addition to entering the metadata, there must be facilities to report errors to downstream users.

Metadata tools are becoming more prevalent. MX2, a tool from Informatica, provides an object-based programming interface for metadata installed with Informatica (Informatica 2006). This program is based on UML. Users can write to metada and access it at both the table and field level. The XML Metadata Interchange specifications are provided by the Object Management Group, which also provides different models to control and manage metadata (OMG 2006). Microsoft provides the Soapsuds tool to assist in compiling client applications that communicate with XML Web services using a technique called “remoting” that enables communication across application domains (Microsoft 2006).

ORGANIZATION

Obviously, this process cannot succeed without organizational support. It may not be so obvious, however, that the organizational structure itself also has a major impact. In organizations where subgroups are autonomous or even competitive, communication between groups may be lacking. We have found that communication links must be defined along the data flow. Users need to know the formal process for reporting errors, and they need to see that these reports are taken seriously and that the errors are corrected. Authority for releasing data and for correcting data must be explicitly defined.

CONCLUSION

We have expanded our original framework into a model that includes metadata tags to help ensure users understand the quality of information in a large, complex system. Conceptually, we can use the metadata structure to store information about (a) the structure of the database, (b) how it is used, and (c) how individual attributes have been managed. In practice, metadata entries need to be automated when data is generated, updated or transformed. Relying on users to initiate entries is not realistic. A supportive metadata tool must be included in the databases, and the metadata for the whole system must be stored in consistent format accessible from all subsystems. In this environment, users can obviously access facts and statistics commonly stored in metadata. However—and more importantly—they can also monitor usage information to help them assess data quality.

REFERENCES

1. Dasu,T. and Johnson, T.(2003). Exploratory Data Mining and Data Cleaning, John Wiley, Hoboken.
2. Hert, C. (2001). Studies of Metadata Creation and Usage. *Proceedings of the Federal Committee on Statistical Methodology Research Conference*, Arlington.
3. Lee,Y.. (1996). Why "Know Why" Knowledge is Useful for Solving Information Quality Problems, *Proceedings of Americas Conference on Information Systems*, 200 - 202.
4. MICROSOFT, http://msdn.microsoft.com/library/en-us/dndotnet/html/introremoting.asp, accessed March, 2006.
5. Morey, R. (1982). Estimating and Improving the Quality of Information in a MIS. *Communications of the ACM*, 25, 5, 337-342.
6. OMG, http://www.omg.org/technology/documents/formal/xmi.htm, , accessed March, 2006.
7. Orr, K..(1998). Data Quality and Systems Theory. *Communication of the ACM*, 41, 2.

8. Rothenberg, J. A Discussion of Data Quality for Verification, Validation, and Certification (VV&C) of Data to be Used in Modeling, Rand Project memorandum PM-709-DMSO, Rand, 1997. http://www.msiac.dmso.mil/vva/Ref_Docs/DataQuality/DataQuality-pr.PDF accessed March 2006.

9. Senne, L., Robbert, MA and Haughton,D. (2005). A Framework for Cleaning Data: Bad Data and Mortgage Decisions. *Proceedings of the CAiSE'05 Workshop*s.

10. Strong, D.,Lee, Y. and Wang,,R. (1997). 10 Potholes in the Road to Information Quality. *Computer*, 30, 8, 38-46.

11. Vassiliadis, P., Bouzeghoub, M.and Quix, C. (1999) Towards Quality-Oriented Data Warehouse Usage and Evolution. *Lecture Notes in Computer Science*. 1626, 164-181.

12. Wand,, Y and Wang, R. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM*, 39, 11, 86-95.