Methodology for Intelligence Database Data Quality

Dr. Arnon S. Rosenthal
Dr. Donna M. Wood
Dr. Eric R. Hughes
The MITRE Corporation
Burlington Road
Bedford, MA 01730, USA
(813) 831-5535
813-835-4661
{arnie, dwood, hughes}@mitre.org

Abstract

Data quality, defined here as fitness for use, is increasingly seen as a serious problem in government and private sector databases. We will survey available techniques, and then describe our own work.

We are adapting general data quality techniques suited to intelligence databases, focusing on an aspect rarely seen in the literature, i.e., helping an intelligence analyst assess individual data records/objects. Our emphasis is on developing solutions to the problem of providing better consumer information on each value used. We provide such information, so the consumer can determine whether the data are good enough for the intended purpose. The primary concern is with individual data items that drive major decisions, where erroneous data have high cost (e.g., human lives). The broad aim is to enable better decisions. A narrower aim is for consumers to trust data when appropriate, thereby reducing the incentives to ignore the data or expend effort on workarounds for data of unknown quality. This paper explains where our approach fits in the spectrum of data quality approaches, and describes a methodology for providing intelligence analysts (consumers) with information needed to guide how they use each data value in making decisions. The methodology encompasses the following aspects:

- *Providing an infrastructure to define, store, and make available quality attributes on various data records/objects*
- *Obtaining values for quality attributes on important data granules*
- *Making the quality attribute values available to users of each data granule (including both humans and queries)*
- *Tracking the impact of providing the quality values, on decision makers and decisions*

Key words - data quality annotations, quality annotation methodology

1. Introduction

Data quality, defined as *fitness for use*, is increasingly seen as a serious problem in government and private sector databases. Our "Managing Data Quality (MDQ)" research is carrying out several investigations in an attempt to develop general techniques suited to intelligence databases. This paper explains where our work fits in the spectrum of data quality approaches, and gives a methodology for providing consumers with information needed to guide how they use each data value in making decisions.

2. Overview of Data Quality Approaches

There are two basic approaches to improving a system's data quality: *defect reduction* and *consumer information.*

Defect reduction efforts receive more attention in the literature. The mainstream of data quality research and products seems driven by data warehousing, enterprise resource planning systems, customer relations, and direct mail. For such efforts, one typically gathers impressions or statistics about the quality of large sets of data (e.g., all customer deliver-to addresses), the benefits of improved quality for each category, and the likely costs of improvement. One then alters the data acquisition and cleaning processes to improve the data values stored within the database. We have done some explorations of this sort (interviewing stakeholders and documenting shared impressions), but this exploratory study used no statistics nor formal methodology. It will not be covered further in this paper.

Consumer information efforts aim to make the existing data more usable, by adding information. One aspect is to better document how one interprets the *meaning* of the data (for example, just how 'Threat' is defined in Army applications or whether a French unit reports distance in meters, feet, or kilometers). Understanding the meaning is particularly important when connecting an automated application, which may not realize that 5 feet is a ridiculous distance for a tank sighting report. Because meaning is covered in the extensive data integration literature, we will not consider it further here.

We focus instead on an aspect rarely seen in the literature, i.e., helping an analyst assess individual data values. We are concerned with individual data items that drive major decisions, where erroneous data have high cost. If there was a single motivating use case, it was avoiding situations like the targeting of the Chinese embassy.

Our task therefore goes beyond the data quality marketplace. Traditionally, data are byproducts of providing goods or services; for our customers, information may be the primary product. Traditional efforts often use data for routine automated transactions; there is little human involvement with each data instance. Errors there are costly in the aggregate (e.g., wasting 10% of a direct mail campaign), but a single wrong data value rarely causes loss of life (or the equivalent in corporate motivation and survival, catastrophic financial loss). In military intelligence settings, a human typically inspects the data before a decision is made.

We aim to provide the consumer with a better picture of an individual item's quality so he can determine whether the data is good enough for the intended purpose. The broad aim is to enable better decisions. A narrower aim is for consumers to perceive the databases' contents as trustworthy, thereby reducing the incentives to ignore the data or expend effort on workarounds.

2.1 Project Setting

The MDQ research is centering its experiments around two operational intelligence databases, their interactions with each other, and a subset of their data producers and consumers. In an intelligence domain, data necessarily give an imperfect picture of the external world. While we are performing informal studies for defect reduction, our efforts have focused on providing consumer information so consumers can more appropriately and confidently employ the data that are available.

For defect reduction, we conferred with data providers, system managers, and some users of the intelligence databases to identify individual data attributes whose quality was perceived as problematic. We selected the quality measures (derived from the literature) that would describe the problem to guide future efforts. The central idea is to allow consumers to see quality values for the data they retrieve. We cast each step in a general light so that the process can be repeated. It is interesting that when compared with a classical data quality view [Wang93], the steps line up fairly exactly, but many of them took a radically different form.

Figure 1 illustrates two variations of the case that we investigate. In both instances, no quality annotations are provided in the intelligence database. When a consumer accesses data directly from the source, as depicted in the top flow of Figure 1, it is obvious that the consumer cannot ascertain quality.

The second case, depicted in the lower flow of Figure 1, presents the problem of consumers accessing data which have been derived from a source database; it is not possible to derive quality that has not been propagated. In this case, even if the producer database contains quality indicators and the consumer is aware that the quality indicators exist, there is no mechanism for the consumer to derive quality values.

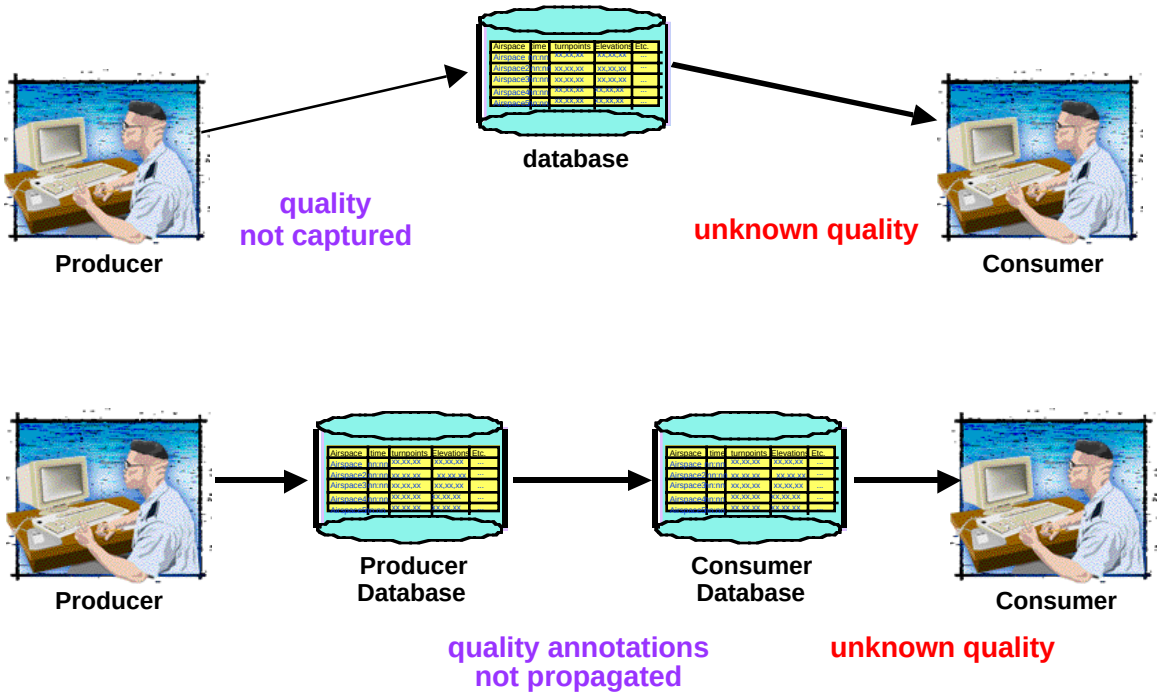

Figure 1. Non-propagation of Data Quality Information

2.2 Overview of Methodology for Providing Consumer Information

These tasks must be accomplished:

- Provide an infrastructure to define, store, and make available quality attributes on various chunks of data
- Obtain values for quality attributes on important data granules
- Make the quality attribute values available to users of each data granule (including both humans and queries)
- Track the impact of providing the quality values, on decision-makers and decisions

The next section provides more detail.

3. Methodology for Providing Consumer Quality Information

The infrastructure's basic task is to allow administrators and power users to attach values of data quality metrics, as annotations, to various chunks (*granules*) of the database, and to make this information available to users of those granules. These concepts are defined below.

A *data quality (DQ) metric* describes the usefulness of some data. Popular examples include measures of accuracy, precision, source, completeness (for sets), and time of observation; others may be derived from these; still others may be collected. Our infrastructure can represent quality metadata on quality values – after all, quality values are data. However, our investigations have not pursued this second order effect. In the future, we may track information *utility* (i.e., benefit of having it), both for ordinary data and for metrics; utility can be a function of quality.

An *annotation* is a triple, (annotated-object, annotation name, value) that is logically attached to some object or other part of a database. An alternative logical view might include some of the annotations as part of the regular database structure. The ordinary interface does not show whether an annotation is physically co-located with the annotated object (e.g., for image metadata) or stored separately.

3.1 Provide an Infrastructure to Define, Store, and Make Available Quality Attributes on Various Chunks of Data

While not strictly part of the methodology, it is interesting to understand the infrastructure provided to support the data quality work. The primary requirement is to be non-intrusive. The infrastructure is able to employ existing data that provides quality information (e.g., dates of capture, error bounds), as well as store separately-provided knowledge, at cell, column, row, and table granularities. A more sophisticated infrastructure could capture knowledge as rules (e.g., If Year=1989 and country=Soviet Union then Accuracy < 0.2). The metrics are made available as separate annotation views, independent of the application tables. The infrastructure provides operations to administer, update, and retrieve data quality metrics whose values are represented as annotations.

Administration splits into definition, view administration, and physical administration. One can define types of annotations (e.g., *accuracy, time-captured, source*) as ordinary data attributes. For each, one supplies a data type, value constraints, and prose describing its meaning. For each attribute that receives that type of annotation, an administrator specifies 1) rules that derive values from contents already in the database, or 2) that explicit storage be allocated. For example, one of the subject intelligence databases contains many attributes that describe how a datum was obtained, or an analyst's estimate of security. These are logically derived into annotations, but need not be physically replicated. View administration controls are shown, by default, as part of the annotation user interface.

The infrastructure maintains the relationship between an annotation value and a chunk in the database, e.g., a table, column, or cell. Annotations are updated as ordinary database data. For read, one can get annotations exactly on a granule, or include super- and/or sub-granules. The infrastructure provides a generic query interface that presents annotations as additional columns of the annotated table. The semantics are those of an ordinary view.

Finally, we note that the infrastructure works for any kinds of annotations one wishes to attach to data values – it is not specific to quality information. Database researchers are making interesting progress on annotating all sorts of information [Delc01,Bird00].

3.2 Obtain Values for Quality Attributes on Important Data Granules

Again, intrusion and extra work is minimized.

The first step is to determine what quality metadata is already provided in the database schema. If possible, we will get providers' agreements to continue supporting these attributes, and to provide fill for them. (One of the subject intelligence databases contains many attributes describing data acquisition and processing, and these provide much of the necessary information.)

Beyond this, we intend to capture wholesale rules that describe all the instances provided by a data feed. This is much cheaper than manually creating each instance.

To plan gathering of further quality information, one works from two sides – need and ease of capture. For need pull, we determine what quality data would make a difference, and be desirable to obtain. As a form of push, we capture quality data that are cheap to get (e.g., time of entry, source).

Builders of data capture software have the option of enforcing data constraints, which sometimes improve data quality. These include value constraints and referential constraints (i.e., that subsidiary data must refer to facilities already in the table). Ease of use must be considered. In Desert Storm, data providers disliked the constraints, and moved much of their content to free-text fields. An alternative is to record constraint violations as annotations for later attention, e.g., to check for alternate spellings.

But even the best capture software cannot provide fill where none is available, nor recheck to determine if an army unit has moved or a building has a new use. Some observations are inherently unreliable (e.g., number of men in enemy unit). For these cases, quality metrics should be provided.

Where part of a record fails the quality checks, we want a means of capturing the good part. (In the past, Airborne Warning and Control System (AWACS) lost considerable data that its data-passing interfaces found somehow faulty; the overall effect on data quality was detrimental.) One approach is to set "bad" values to null, with an annotation holding the suggested value so it is not lost. Automated applications will need to be null-aware, i.e., to behave correctly with null data.

3.3 Make the Quality Attributes Values Available to Users of Each Data Granule (Including Both Humans and Queries)

We explore two means of providing quality values to users -- non-intrusively. Figure 2 illustrates our concept. The producer provides quality annotations which are captured by our tool in a separate but related

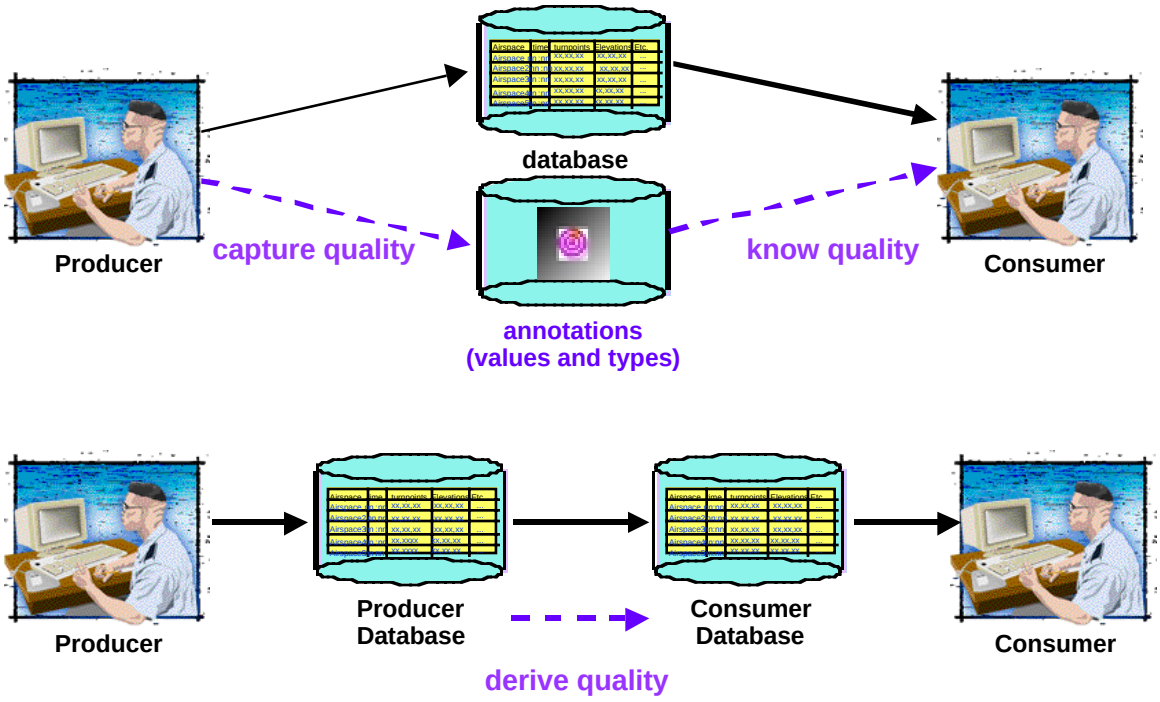

Figure 2. MDQ Proposed Methodology

database. These annotations are then propagated either directly to the consumer, or to the consumer database. We note that the consumer needs control over whether screen space is devoted to these extra columns when displaying the results of database queries. In both cases, we have provided a very basic implementation. We also modified two existing user interfaces: one for querying, and the other for map display. In both cases, once the implementers understood the user interface's code, a few hundred new lines sufficed. We are convinced that a fuller implementation need not be very difficult. Generic interfaces for annotations should be provided for the most common forms, i.e., relational (which we have completed) and Extensible Markup Language (XML).

3.4 Track the Impact of Providing the Quality Values on Decision Makers and Decisions

We anticipate that it will be very hard to track the impact of quality metadata on user decisions. Several techniques seem natural. For now, we lean toward using only the first, which is least intrusive:

- Our interfaces make display of quality values optional, generating different queries based on what quality values the user wants retrieved. We can track whether the users include the quality data in their displays (though not its influence on their decisions).
- Survey users about what they use and how valuable it is.
- Provide a box for rating the utility of metadata, as part of the user interface. (For example, Amazon.com lets users rate the utility of feedback from a reviewer)

4. Conclusions and Future Work

We have shown how intelligence database systems can be extended to manage data quality annotations on base data. The critical next step is to obtain real users, and improve the system based on their feedback. We have also identified a future research direction: we hope to provide the means for creating and managing tailored views (comprising both query and display) for communities of interest (COIs), and for adding data quality capabilities to these views. See Figure 3. We aim to reduce the cost and delays in producing and maintaining tailored interfaces, thereby enabling better ones to be provided. To do so, we will build a componentized view capability that can address both query and display aspects of an interface. We will show how this view capability will allow COIs to form dynamically, collaborate through a view, and update data via the view. We intend to investigate mechanisms for treating feedback on quality annotations, and will explore techniques to dynamically choose source information with quality annotations considered.

References

[Bird00] S. Bird, P. Buneman, W-C Tan, "Towards a query language for annotation graphs" *Conference on Language Resources and Evaluation 2000.*

[Delc01] L.Delcambre, D. Maier, et. al., "Bundles in Captivity: An Application of Superimposed Information," *IEEE International Conference on Data Engineering 2001.*

[Wang93] R. Wang, H. Kon, S. Madnick, "Data Quality Requirements Analysis and Modeling," *IEEE International Conference on Data Engineering 1993.*

Appendix A Comparison With Other Methodologies

To see how military intelligence problems are different, we compare with a methodology motivated by finance and industrial databases [Wang93]. The points of difference are:

- Intelligence agencies often have individual data values that drive important decisions, and are evaluated by humans, rather than by tool.
- Some methodologies suggest filtering out data that doesn't meet standards. In many intelligence applications, one uses the best data available – more cautiously if it is not very good.
- Aging is a major problem with much of intelligence data.
- Coverage can be *very* sparse, and too costly to increase.
- Other methodologies require the data administrator to estimate quality, since data capture is by clerks. With professional information analysts, one may often get good estimates
- The MDQ research is investigating the improvement of an existing system, not the design of a new system from scratch.
- The MDQ methodology has no need to treat subjective and objective metrics differently.

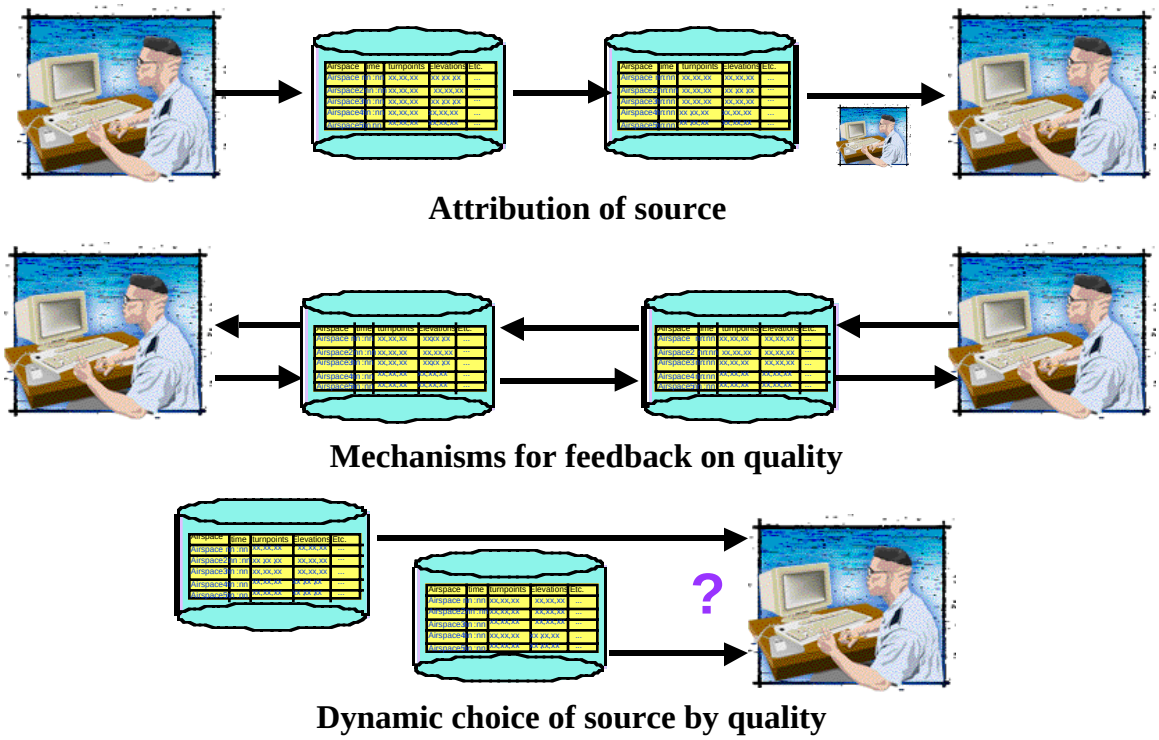

Figure 3. Future Plans

Author Biographies

Dr. Arnon S. Rosenthal (781-271-7577) is a Principal Scientist in the Center for Integrated Intelligence Systems (CIIS) at the MITRE Corporation in Bedford, MA. Dr. Rosenthal is well known in the database field and has recently focused on data integration, system administration, security, and data quality research.

Dr. Donna M. Wood (813-831-5535) is a Lead Scientist in the CIIS at the MITRE Corporation in Tampa, FL. Dr. Wood has spent approximately 25 years supporting the United States Air Force in information system engineering and integration efforts, much of it focusing on database and data issues. In addition to contributing to this research, she is currently supporting the United States Special Operations Command (USSOCOM) in developing long-range Information and Information Technology (IT) Management strategies.

Dr. Eric R. Hughes (781-271-7486) is an Associate Department Head in the CIIS at the MITRE Corporation in Bedford, MA. Dr. Hughes has applied his interests in object databases and distributed systems to Air Force and intelligence applications in his 5 years at MITRE.