Error Browsing and Mediation: Interoperability Regarding Data Error

Henry B. Kon
Michael D. Siegel

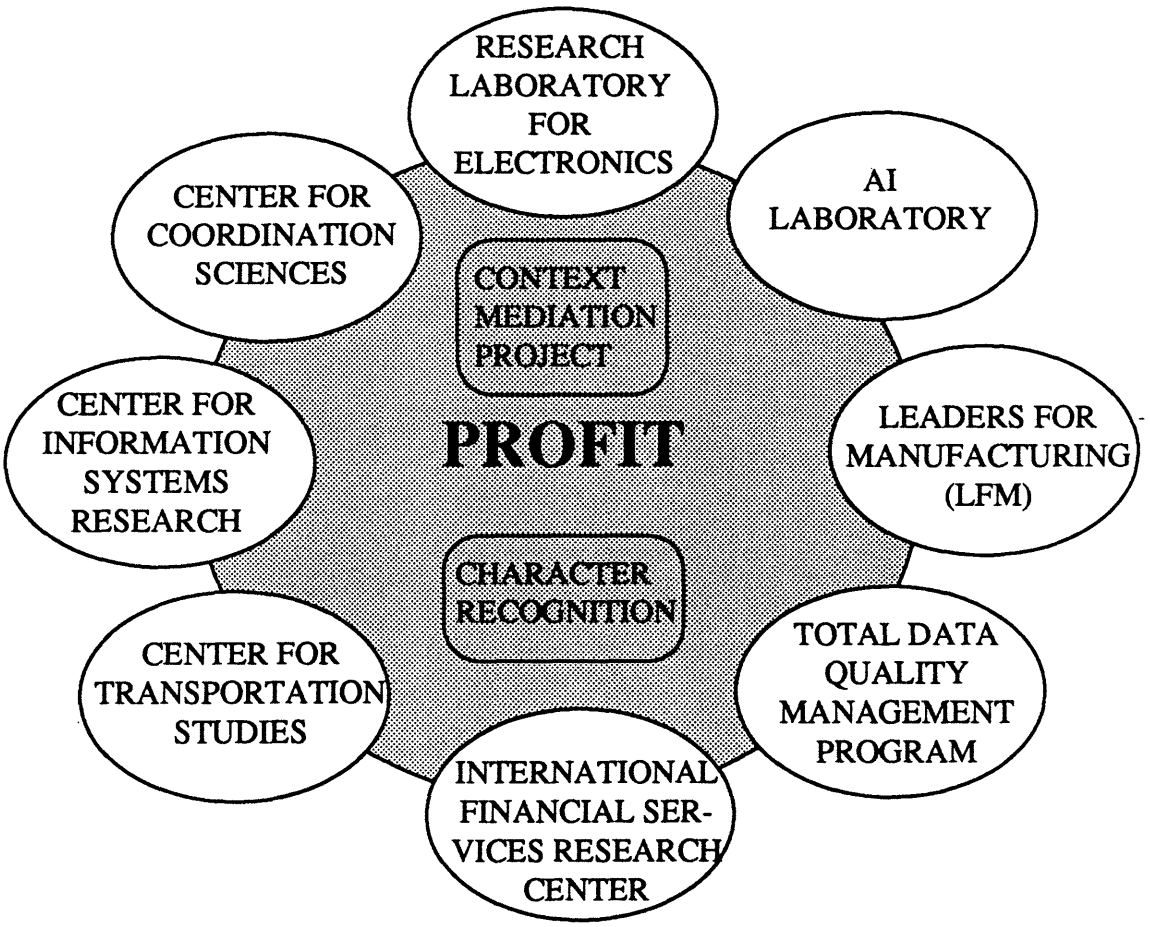
WP #3776 July 1994
PROFIT #94-15

Productivity From Information Technology
"PROFIT" Research Initiative
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
(617)253-8584
Fax: (617)258-7579

 The research described herein has been supported (in whole or in part) by the Productivity From Information Technology (PROFIT) Research Initiative at MIT.

Productivity From Information Technology (PROFIT)

The Productivity From Information Technology (PROFIT) Initiative was established on October 23, 1992 by MIT President Charles Vest and Provost Mark Wrighton "to study the use of information technology in both the private and public sectors and to enhance productivity in areas ranging from finance to transportation, and from manufacturing to telecommunications." At the time of its inception, PROFIT took over the Composite Information Systems Laboratory and Handwritten Character Recognition Laboratory. These two laboratories are now involved in research related to context mediation and imaging respectively.



In addition, PROFIT has undertaken joint efforts with a number of research centers, laboratories, and programs at MIT, and the results of these efforts are documented in Discussion Papers published by PROFIT and/or the collaborating MIT entity.

Correspondence can be addressed to:

The "PROFIT" Initiative
Room E53-310, MIT
50 Memorial Drive
Cambridge, MA 02142-1247
Tel: (617) 253-8584
Fax: (617) 258-7579
E-Mail: profit@mit.edu

Error Browsing and Mediation: Interoperability Regarding Data Error

Henry B. Kon
Michael D. Siegel

July, 1994

hkon@mit.edu msiegel@mit.edu

MIT Sloan School of Management
Information Technologies Group
30 Wadsworth St., E53-320
Cambridge, MA 02139

Overview: Our research goals involve development of methodologies and systems to support administration and sharing of errored data (e.g., data having incompleteness, inaccuracy, and invalid syntax). Data sources are assumed to have non-trivial degree of error. Data receivers are assumed to have differing sensitivity to various forms of error. Browsing involves measurement of error. Mediation involves run-time management of the source-receiver "error fit".

In this extended abstract we provide a foundation for error definition and measurement, and discuss their role in browsing and mediation. Included are: (1) a classification scheme for error definition as *syntactic error* and *semantic error* types, (2) a theoretical basis for relating semantic error to data meaning, (3) an outline of three general approaches to error measurement, and (4) an overview of browsing and mediation. It is our contention that this approach to data error, though complementary to existing approaches, is novel.

1. Introduction

Error in data has been measured as high as 50% in a variety of application contexts, including accounts receivable, inventory, financial, military personnel, and criminal justice [3, 11, 15, 17, 18]. Two explanations may help account for this. First, database applications rarely reflect the exact needs and capabilities of their users so, for example, attributes may exist in the database schema may no longer be used by an application. Second, not all application contexts require error-free data. Different applications have different error requirements.

Two error-related strategies exist: reduction and measurement. Reduction is achieved by correcting an errored data set or the process that created it. Measurement involves reporting error so that receivers can be aware of error in the data they use. Reduction is of limited applicability however, for two primary reasons. First, it presumes control over the data by those concerned with error. While viable for local data, this does not apply where source and receiver operate autonomously. Data providers may be unwilling to improve data, data may be unrepairable due to a historic or specialized nature, and data can not always "keep up" with a rapidly changing reality. Second, we may not want to eliminate error entirely. Because error reduction is not cost-free, an economically optimal state in data processing will involve some degree of error. There are diminishing returns and inefficiencies in achieving zero-defect data.

In this abstract, we introduce the browsing and mediation approach to error in structured data. Rather than attempt to reduce error (directly), the focus is on error measurement via software systems which both produce and utilize error measurements. Our primary objective in these pages is to provide a definition for error and to outline approaches to error measurement. We believe that the issues identified are fundamental for progress in this area, and that the approach presented is novel.

Management of error in data is not a new research concept (e.g., integrity constraints, querying incomplete data [10]). Error is also analyzed within particular application domains (e.g., biases in accounting reports [11], under counts in census data [4]). These approaches are focused not on general error measurement, however, but on particular forms of error detection and inferencing about error.

Our approach differs on several counts. First, we seek a comprehensive and domain-independent concept of error. Second, we are interested in formal error measurement and representation of receiver sensitivities. Third, we aim to develop browsers and mediators to sit between sources and receivers, allowing both human interaction with data error during browsing, as well as automatic run-time mediation. A high-level architecture is shown in Figure 1 below.

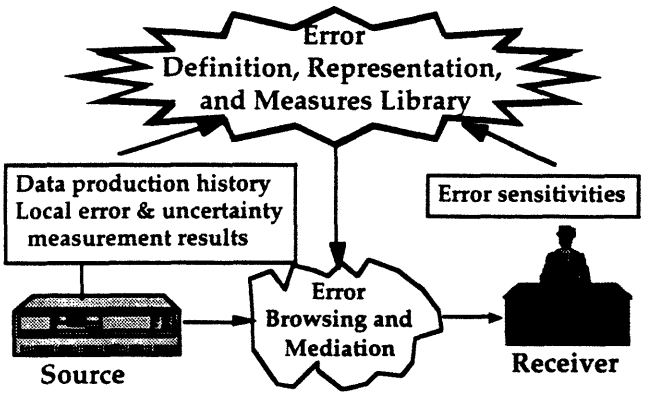


Figure 1: Error browsing and mediation: a source-receiver architecture

As a foundation, we adopt portions of Bunge's mathematical approach to semantics [6, 7], ontology [8], and epistemology [9]. From it, we borrow concepts to relate data error, data meaning, and data collection. Below is a diagram of research components relevant to this project. Roughly, the components depend on those underneath them, where browsing and mediation are the final goals. For example, browsing (7) requires measurement (4), which in turn requires definition (1).

(8) mediation	managing source-receiver error fit within autonomous, heterogeneous data federation. Sensitivities and measures information is shared. Measurements beyond error thresholds signal software alarms.
(7) browsing	interactive software for error measurement and reporting, data/error visualization. Support for error (data process) monitoring over time.
(6) sensitivity	explication and representation of receiver error definitions and sensitivities
(5) representation	how to attach measures to a data model and data collection history to data.
(4) measurement	how, operationally speaking, to perform error level assessment. Specification of measurement algorithms and methods.
(3) collection	tagging data with its *collection method*, e.g., source, aging, derivation method, etc.
(2) measures	how to quantitatively describe error levels via numeric scales
(1) definition	what is an error. How to identify distinct error types
data definition (meaning & representation of data)	

Table 1: Error browsing and mediation research components (items 1-8)

This abstract concerns definition and measurement (items 1 and 4), and browsing and mediation (items 7 and 8). In Section 2, we provide a scheme for defining error and classifying error types. Section 3 covers error measurement. Section 4 provides an overview of error browsing and mediation. In Section 5, we summarize and discuss future directions for this research.

2. Error Definition

In this section we propose a framework for error definition. First, we critique the concept of data error as it exists in the literature. Next, we provide a theoretical foundation for answering "what is an error?". Lastly, we provide an error classification scheme based on semantic and syntactic error types.

2.1. Previous work

The terms 'accuracy' and 'completeness' seem to encompass all possible types of error: data can be either present and incorrect (inaccurate), or "the data" may not be present at all (incomplete). While comprehensive, these are too abstract to support measurement. 'Accuracy' has different connotations for categorical versus quantitative data. 'Incomplete' might be a characteristic of a database, table, column, record, or cell.

Two tables below contain example data, based on a university alumni application. Table 2 contains a listing of Alumni who do fundraising for a school, including their Name, State of residence, and Year of graduation. Table 3 contains Name and State in common with Table 2, and lists also the Population of the state.

Fundraising-alumni		
Name	**State**	**Year**
Pete Wilson	Massachusetts	1969
Jennie Wales	Rhode Island	1990
Frank Ames	New York	1974

Table 2: Example data: university alumni

Fundraising-alumni2		
Name	**State**	**Population**
Pete Wilson	Massachusetts	5,998,000
Jennie Wales	Rhode Island	1,005,000
Frank Ames	New York	18,119,000

Table 3: Non-normalized table.

To analyze error, researchers typically view a database as a collection (stream) of data records (items) in which error is either present or is absent [2, 3, 17, 18, 19, 20]. In these works, error is defined as a binary measure, which applies at the record level of analysis. The error measure for a table, i.e. a collection of records, is based on percentage-in-error. Below is a list of key shortcomings in such research.

- (1) **Misleading measures** Suppose that Table 2 had two incorrect State values (column 2) and zero incorrect Years (column 3). With the record as the unit of error analysis, Table 2 would be 33% correct since two out of three records have an error. Applications interested only in Year, however, would find these error measures misleading because all Year values are correct.
- (2) **Meaning ambiguity** Types of error associated with attribute values (Population) should be different from those associated with object identifiers (Name). The State column in Table 3 above serves as both an attribute and an object identifier. A robust error characterization scheme must distinguish these two roles.
- (3) **Ambiguous incompletes** It is not always clear what "incomplete" data is. The relationship between incompleteness and missing attribute values is not explicit. Incompleteness is ignored entirely in some cases because missing records do not "exist" to be incomplete.
- (4) **No degrees of error** There is no degree of error at the value-level. A numeric value (Population) should have degree of error rather than simply presence or absence of error.
- (5) **Ambiguous Inconsistency** Some researchers describe data 'inconsistency' as a form of error, without raising the issue of which (if not both) of two inconsistent data are actually in error.
- (6) **No syntax error** Untrue statements are not differentiated from syntactically invalid data - another important form of data error.
- (7) **Not of a practical measurement orientation** For much of the error-related research, the result is primarily a theoretical view of a particular form of error. A general and practical proposal for error definition and measurement has not been found in our literature review.

While appropriate for the analyses developed, these various notions of data error are seen to be inadequate and semantically ambiguous. In Section 5, we describe how our error framework addresses these issues.

2.2. What is an error?

The fundamental question remains: What is an error? In this section, we analyze the semantic concept of truth in data, i.e., correspondence between data and the real world. The key outcome is a statement on the relationship between truth in data and data meaning.

Consistent with Bunge's semantics, we assert that truth in data is relative to data meaning (e.g., the schema). Bunge summarizes the relationship: "truth depends on meaning, but not conversely: one and the same statement may be applied now one truth value, now another, without any change in meaning" [7]. Consider testing semantic error, i.e., truth in a symbol, say "JUL-69". Such testing would require one to know about the data meaning: (1) does the symbol represent a thing or an attribute value, (2) if it is an attribute value - for what attribute, and what thing is being described, (3) what units and scale are used, and (4) how are we to read the syntax. Only given meaning of data can one assess its truth.

In some cases, the "same data" can be generated independently by two different *collection methods*, e.g., two ways to measure 'lifetime-contribution' of alumnus X. The two resulting values may not agree. Then, two data values having the same apparent meaning (lifetime-contribution of X), would be both correct and conflicting - a paradox. To solve this we recognize that assumptions about how we collect (e.g., generate, obtain, measure, calculate, judge) a datum are actually embedded in the concept of that datum.

On this basis, we extend the concept of the attribute's meaning from 'lifetime-contribution' to 'lifetime-contribution as obtained by collection method Y'. To the meaning of a datum we append its collection method. This represents an extension from the traditional schematic view of data meaning (e.g., entity-attribute-relationship) to include the procedure used to obtain the data. As a result, these "conflicting yet correct" statements will not actually conflict.[1] We define data collection to include raw empirical observation, as well as subsequent propagation and processing by human or computer. The various collection methods constituting a datum's derivation has pedigree (lineage and breed) implications for the datum.

Sometimes a single data collection method is not self-consistent, i.e., results vary each time the method is applied. This is true of census methodologies for population count. Given constant population, two executions of a census will likely result in two unequal population values. Are either or both of the data therefore incorrect? Under one philosophical interpretation, such data are inaccurate since they are only estimates [7]. However, most people would not want to characterize all population data as being in error on this basis. Generally, a population figure is considered accurate if either: (1) it is consistent with a published data collection result (e.g., the 1990 U.S. Census Bureau report) or (2) it results from data collection performed in conformance with a specification for that collection method.

As an example of these concepts, we have been working with a commercial data service that provides socio-economic data. As depicted in Figure 2 below, a government source (A) performs a census method, the result of which is passed to the data service (B) who, in turn, sells the data to the end consumer (C). A *data supply chain* is formed by the data flow from A to B to C. Each link in the data supply chain is a instance of data collection. A's collection method is a census methodology. B's collection method consists of copying and distributing data collected from A, etc.

Though concerned about error levels, B does not scrutinize (e.g., verify) government data because there are no means for doing so. Some data are thus taken "at face value" - B seeing itself as a pass-through service and considering its output data to be error-free as long as it is consistent with results provided by A (i.e., as long as B performs its collection method correctly, independent of whether the data reflects the true population). C, however, may not be satisfied with the output of A as the final reference point for truth in the data. C may wish to look further up the supply chain - to question A's data and see if it does, in fact, reflect the true population.

---

[1] This approach is consistent with operationist semantics in which the meaning of a statement is equated with the procedure used to obtain it. It is also consistent with the nominalist stance in which intrinsic properties of things can not be known, e.g., the true speed of a comet, but only construed via perceptual or scientific means.

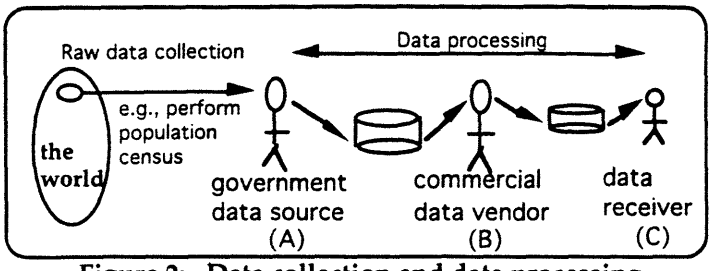


Figure 2: Data collection and data processing

We define a derived datum's collection method as the concatenation of the data collection methods along its supply chain. This provides an explicit frame of reference from which to assert the occurrence of an error. As data propagates through a data supply chain, "error guarantees" may be nullified, passed through, or introduced by data handling agents. A data service may wish to guarantee clean copies of the data, it may wish to clean up, verify, and guarantee data, or "take it as it is" may be its policy.

In answering "what is an error?", we have firstly concluded that data error depends on data meaning. Second, we have extended the concept of meaning to include the collection method. Links in the data supply chain are instances of data collections, together constituting the data production history of a datum.

2.3. An error definition framework: semantic and syntactic error

We propose in this section a framework consisting of two error types: semantic error (regarding truth) and syntactic error (regarding format). Three kinds of semantic error are identified. Consistent with Bunge, we treat data as symbols which are interpreted to derive statements. We distinguish data from statements because only data have syntax error and only statements concern the semantic concept of truth .

2.3.1. Semantic error: true and untrue statements

A statement, according to Bunge, is the result of applying a predicate function (e.g., State-of-residence) to a set of predicate arguments (e.g., Person, State). A predicate instantiated with arguments constitutes a statement. An example statement as interpreted from Table 2 is: 'State-of-residence (Frank Ames, New York)'. Bunge provides a framework for possible types of predicate arguments. Primary among these are a thing (object identifier) and an attribute (value) of that thing regarding that predicate. Other common argument types include a reference frame (e.g., spatio-temporal: date/time/place), as well as unit ($ vs. £) and scale (100's vs. 1000's). An example statement containing these argument types is: 'Income (Frank Ames, 41, 1993, $, 1000's)' which states that Frank Ames' salary was $41,000 in 1994.

These two types of statements are distinguished based on whether their underlying predicate is of an attribution or a classification nature. Different kinds of error will be associated with each. Example attribute and class statements are shown below in Table 4. An *attribute statement* associates a value with some attribute of some thing. The State-of-residence and Salary predicates below give rise to attribute statements. A *class statement*, on the other hand, assigns a thing to a category - it has only a single argument (an object identifier) and states that the denoted object is a member of the class named by the predicate.

Attribute statement	State-of-residence (Frank Ames, New York) - a categorical attribute Salary (Frank Ames, 90000) - a quantitative attribute
Class statement	New-Yorker (Frank Ames) Well-paid-alumnus (Frank Ames)

Table 4: Examples of attribute and class statements

An error in a statement might occur anywhere within that statement - in the predicate or any of its arguments. To define semantic error we focus our attention on the object identifier (e.g., Frank Ames) and the attribute value (e.g., 90000). To measure error in an attribute statement, we assess the *accuracy* of the attribute value relative to the object it characterizes. Later, we discuss how this can be achieved, which is an issue of error measurement and not error definition.

The class statement differs from the attribute statement as it tells us that a thing is an instance of a class. Whereas an attribute statement refers to a single thing, a class statement (also) refers to a population,

i.e., members of the class. A collection of class statements (based on the same class predicate, e.g., Fundraising-alumni) represents the extent of a class - its members, size, total-income, etc. For class statements, we are concerned with issues of incompleteness and mis-classification. The symbols should be complete (denote all class members) and have no mis-classifieds (denote none other than class members). As summarized in Figure 3 below, we associate two semantic error types with the class statement, and one with the attribute statement.

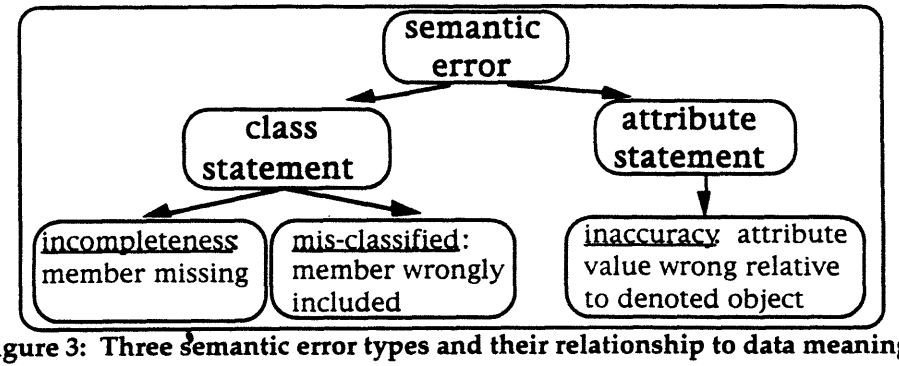


Figure 3: Three semantic error types and their relationship to data meaning

In summary, our semantic construction of error identifies incompleteness, mis-classification, and (attribute value) inaccuracy as three fundamental semantic error types. These are derived from the class and attribute statement types, focusing on error in the object identifier and attribute value predicate argument types. We believe that class and attribute statements reflect key aspects of data meaning, and therefore data truth. Our error types provide a large degree of application- and data model- independence. While the terms inaccuracy and incompleteness have appeared often in the literature, the concept of mis-classification has not been previously identified as a distinct form or data error.

2.3.2. Syntax error: unreadable data

A syntax specification defines correct representations for data symbols. Data having syntax error can not be read, for example, '[July IV, @#$]'. Such data are therefore not mappable to a statement and have no relevance to the semantic concept of truth. We are concerned with syntax at the data cell level, i.e., strings, sub-strings, and letters within a cell. Forms of syntax error include invalid string length, missing delimiters, and invalid characters. There may of course be multiple valid syntax for a given data type.

As we did with semantic error, to define syntax error we would like to identify various syntax error types. It appears, however, that there exists no research precedent for classifying syntax error types. While Bunge offers a framework for semantic truth, he has none to help us categorize aspects of syntax. Though work in database interoperability considers schematic heterogeneity, there is also nothing in this literature on specifying syntax (e.g.,[5, 13, 14]). With no foundation for classifying syntax error types, we will address this from an engineering standpoint - in the browser, where BNF syntax specifications will be used to check syntax [1]. (Because syntax may be trivially convertible among systems, consistency within any syntax scheme may be more important than conformity to a particular one. Syntax measures may reflect both of these concepts.)

We have identified shortcomings in previous definitions of error, answered "what is an error?", proposed syntactic and semantic error types as a classification scheme, and specified three types of semantic error. Next, we discuss how measurement can occur.

3. Error Measurement

Error measurement is a process performed to assess error levels in actual data. Given a database, how do we assess its error? We assume for the present discussion that error measures and representations have been defined (Table 1). Error measures will be specified for each error check being done. These error measures apply to both the individual level (e.g., single data value) and collection level (e.g., entire database)

of data aggregation. Given that syntax error detection is largely an automatic process, we focus here on measurement of semantic error, i.e., truth.

Bunge defines assessment of truth as the application of a truth testing procedure [7]. Agmon proposes constraint checkers as a measurement tool [2]. Motro [18] imagined a hypothetical perfect *reference data* set, against which to compare data in question. Ballou identifies timeliness as contributing to degradation of data [3].

Based on these three examples, we identify three semantic error measurement types: internal, external, and process, each embodying a different approach to truth assessment and testing. Internal involves stand-alone testing of data using constraints as checkers. External involves comparing data against the real world (or against an independent reference data source). Process is based on inference about degree of error based on the data's collection method. (These are akin to the coherence, correspondence, and model-theoretic concepts of truth identified in [7].) These three measurement types are depicted in Figure 5 below, and are described next. As shown, the net result of error measurement is error metadata.

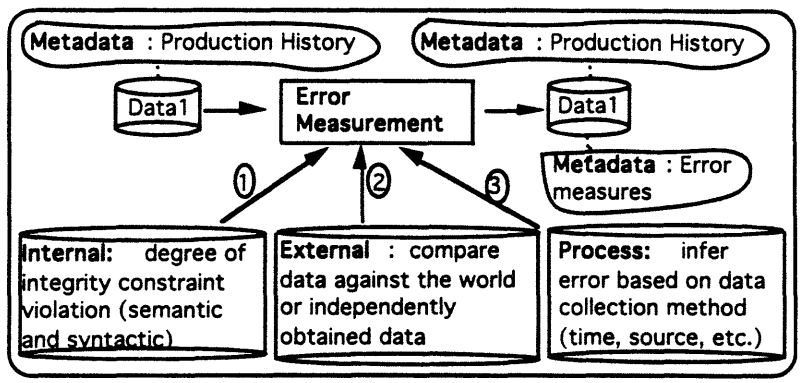


Figure 5: Three measurement types: internal, external, and process

Arrow #1 (Internal): Semantic integrity constraints delimit valid data values and relationships based on known laws about the world. Though typically used for error prevention and detection, 'degree of constraint violation' constitutes an inexpensive error measure, e.g., 'percentage of attribute values out of range or percentage of records logically invalid'. Constraint-based measures provide a lower bound on the actual error level because data can be logically feasible yet still incorrect. The same technique applies to syntactic constraints which are associated with data types: floating point, alphanumeric, date, address, etc.

Arrow #2 (External):

(External - inter-database validation): Data can be validated by comparing against other data. This approach applies whenever data are inter-related by known algorithms and the external data is "reasonably independent" or known to be factual. (Otherwise we might consider this an internal consistency measure of a distributed database.) Inter-database relationships across which comparison/verification can be performed might include redundancy (same information stored in two places), arithmetic (x+y=z), or logical (foreign keys). By comparing data with other databases, we can identify data as suspicious, determine exact error levels if the second database were assumed error-free, or estimate error via a stochastic model of the joint probability of error in the first and second databases.

(External - real world verification): This approach is similar in spirit to inter-database validation, but applies when the data can be verified empirically. Such verification might involve reweighing an object or making a phone call to verify a price. Sampling will be required for large databases where verification is labor intensive. Sampling may be required in "both directions" - to test mis-classification and inaccuracy we sample data and test against the world (or other data). To test completeness, we sample entities in the world and see if they are listed in the database.

Arrow #3 (Process): A datum's production history includes how it was born (e.g., collected as raw data) and various aspects of its processing [16]. Process-based measures involve assessing error based on the data collection method - how, when, by whom, with what, etc. - data was generated and processed. Here are three

examples of (user-specified) process-based measures - regarding data age, source, and collection method respectively (three key ingredients of a data production history):
(1) (aging/timeliness) A 5 hour old stock quote has a wider range of possible values than one 2 seconds old.
(2) (source) Data from the Wall Street Journal is more accurate than from The Enquirer tabloid.
(3) (collection method) Data generated in a job interview is more accurate than from a marketing survey.

Uncertainty-based representation schemes must accompany such heuristic inference.

4. Error Browsing and Error Mediation, an overview

Application of our error framework is in browsing and mediation. The ultimate goal underlying these two is to facilitate and automate error measurement. Browsing allows a user assessment of data error via data browsing and interactive error measurement. Mediation focuses on managing dynamically the source-receiver error fit. In this section we give a brief overview of each. Though our prototype system is in a design stage, we provide here a summary of anticipated functionality. The high-level architecture for browsing and mediation was given in Figure 1.

4.1. Browsing

Browsing is a general approach to interaction with data in which users can easily scan the data, data dictionaries, metadata, etc. [16]. The error browser focuses on a user interface for error understanding and assessment. Input to the browser consists of knowledge about what errors are of interest to a receiver, and about how they are to be measured, e.g., % records violating semantic constraint #1, % values inaccurate in column X, completeness of class Y based on inter-database comparison with database Z, etc. Knowledge will also be input regarding data collection methods, e.g., specifying source, time, data generation and derivation. For a source, the data collection method is inherent in the data (e.g., the data is six days old from data source B). The receiver specifies these as requirements (e.g., "data may not be more than four days old").

Browser output consists of the results of error measurement as performed on actual data sets. These output may be in textual or graphical form, for human or computer consumption. For each error measure, a numeric error level and examples of data found to be in error will be provided.

Sometimes, basic aspects of data meaning are not clear, e.g., in a legacy system or in data void of a robust database schema [12]. Where meaning is not explicit, null data cells and replicated values (e.g., within a column) may constitute errors (nulls or replicates in a unique key column, for example). To address these, the browser will provide data value distributions of both data values and of data replications. Data distributions can also make data "blips" obvious - these are data which are not otherwise suspected of being in error, but are conspicuous because they differ significantly from other data in their vicinity. Blips may be detected in a list of alumni salaries (e.g., $1,200/yr looks suspicious), in temperature readings across a region, or in a time-series of annual donation totals by a single alumnus over the years. Blips are merely suggestions about possible error in data that have not undergone an official test. The errors associated with nulls, replicates, and blips (assuming them to be syntactically valid data) are all semantic errors.

We have discussed the browser, with its inputs from both source and receiver, as well as the role of data distributions in error detection.

4.2. Mediation

Mediators are software components that address heterogeneities among federations of autonomous data and application systems [21]. Regarding error, data sources (data sets) are heterogeneous both in terms of the error level and in terms of the data collection method. Data receivers (application contexts) are heterogeneous in terms of the error measures of interest, and in terms of error sensitivities. Error mediation enables sharing of such information in both directions. The error mediator will dynamically assess and manage the fit between source and receiver. Based on error measurement results, the mediator can detect source-receiver conflicts in which the data appears to have too many errors for the receiver. Given error measurements and receiver sensitivities, error thresholds can trigger alarms when various irregularities or error threshold violations are detected. A "RED/unacceptable YELLOW/caution GREEN/ok" rating system should be the appropriate degree of granularity for error-related decision making (i.e., "don't use vs. investigate further vs. data ok"). In addition to detecting conflicts, the mediator may attempt to repair

aspects of data error where possible (e.g., supplement an incomplete database with a second source, correct syntax violations, etc.). The mediator may act automatically to reject data or it may signal an application warning.

5. Conclusion

Conventional systems exchange only application data. With the shift over time to browsable data repositories incorporating hundreds or millions of data sets, computer-assisted error measurement will become required functionality. Browser functionality may be used by end users for database shopping. Mediation may be most appropriate for database monitoring and control by data administrators over time.

In this abstract we have answered the question: "what is an error?". Our syntactic-semantic distinction, and the truth-meaning relationship, offer a useful way to view the issue. We provided a general error definition framework - based on the simple but comprehensive semantic-syntactic error classification scheme. The data collection method, seen as a facet of data meaning, offers a foundation for testing semantic truth. Internal, external, and process are three different schemes for error measurement. Error browsing and mediation are new concepts to the research literature.

This error framework addresses several of the major concerns presented in points 1 through 7 in Section 2.1. We briefly summarize why this is so. Most importantly, we link error measures to concepts of meaning. Our measures are thus semantically precise so that no misleading or ambiguous measures can exist. Syntax error is an explicit and measurable aspect of error in this system. Practical error measurement, driven by real world data and systems, are a primary goal of our research.

We expect this to be a research process in which, over time, different pieces of analytic machinery fit into a common error browsing and mediation framework. Uncertainty and decision-theoretic schemes may facilitate inference (e.g., statistics, theory of evidence, decision theory, fuzzy sets). Existing software tools may also be useful (e.g., data visualization, metadata representations). We are currently developing a probabilistic error propagation calculus which calculates semantic error in query output as a function of semantic error in query input. Several research issues remain: definition and representation of a data collection method, explication of error measures (numeric scales) and receiver sensitivities, and error representation along with its requisite semantic to structural mappings. These are issues for future research. The current framework serves as a starting point from which error browsers and mediators can be considered.

6. References

[1] Aasa, A., Petersson, K., & Synek, D. Concrete syntax for Data Objects in Functional Languages. *Proceedings of the 1988 ACM Conference on LISP and Functional Programming*: 96-105, Snowbird, Utah, 1988.

[2] Agmon, A., & Ahituv, N. Assessing Data Reliability in an Information System. *Journal of Management Information Systems*, 1987.

[3] Ballou, D. P., & Pazer, H. L. Modeling Data and Process Quality in Multi-input, Multi-output Information Systems. *Management Science, 31*, 2: 150-162, 1985.

[4] Barabba, V. P. Proceedings of the 1980 Conference on Census Undercount., Arlington, Va., 1980.

[5] Batini, C., Lenzirini, M., & Navathe, S. A comparative analysis of methodologies for database schema integration. *ACM Computing Survey, 18*, 4: 323 - 364, 1986.

[6] Bunge, M. Semantics I: Sense and Reference. D. Reidel Publishing Company, Boston, 1974.

[7] Bunge, M. Semantics II: Interpretation and Truth. D. Reidel Publishing Company, Boston, 1974.

[8] Bunge, M. Ontology I: The Furniture of the World. D. Reidel Publishing Company, Boston, 1977.

[9] Bunge, M. Epistemology & Methodology I: Exploring the World. D. Reidel Publishing Company, Boston, 1983.

[10] Imielinski, T., & Lipski, W. Incomplete Information in Relational Databases. *Transactions on Database Systems*, 1984.

[11] Johnson, J. R., Leitch, R. A., & Neter, J. Characteristics of Errors in Accounts Receivable and Inventory Audits. *Accounting Review, 56*, April: 270-293, 1981.

[12] Kent, W. Limitations of Record-Based Information Models. *ACM Transactions on Database Systems*, 4, 4, 1979.

[13] Kim, W., & Seo, J. Classifying Schematic Data Heterogeneity in Multidatabase Systems. *IEEE Computer, 24,* 12: 12-18, 1991.

[14] Krishnamurthy, R., Litwin, W., & Kent, W. Language Features for Interoperability of Databases with Schematic Discrepancies. *Proceedings of the ACM SIGMOD Int'l Conf. on the Mgt. of Data,* Denver, 1991.

[15] Laudon, K. C. Data Quality and Due Process in Large Interorganizational Record Systems. *Communications of the ACM, 29,* 1: 4-11, 1986.

[16] McCarthy, J. L. Metadata Management for Large Statistical Databases. *VLDB*: 234-243, Mexico City, Mexico, 1982.

[17] Morey, R. C. Estimating and Improving the Quality of Information in the MIS. *Communications of the ACM, 25,* May: 337-342, 1982.

[18] Motro, A. Integrity = Validity + Completeness. *ACM Transactions on Database Systems, 14,* 4: 480-502, 1989.

[19] Paradice, D. B., & Fuerst, W. L. An MIS Data Quality Methodology Based on Optimal Error Detection. *Journal of Information Systems, Spring*: 48-65, 1991.

[20] West, M., & Winkler, R. L. Data Base Error Trapping and Prediction. *Journal of the American Statistical Association, 86,* 416: 987-996, 1991.

[21] Wiederhold, G. Mediators in the Architecture of Future Information Systems. *IEEE Computer,* March, 1992.