Published in the *Ninth International Conference of Data Engineering*
Vienna, Austria, April 1993

Data Quality Requirements Analysis and Modeling

December 1992 TDQM-92-03

Richard Y. Wang
Henry B. Kon
Stuart E. Madnick

Total Data Quality Management (TDQM) Research Program
Room E53-320
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
617-253-2656
Fax: 617-253-3321

Acknowledgments: Work reported herein has been supported, in part, by MITís Total Data Quality Management (TDQM) Research Program, MITís International Financial Services Research Center (IFSRC), Fujitsu Personal Systems, Inc. and Bull-HN. The authors wish to thank Gretchen Fisher for helping prepare this manuscript.

Data Quality Requirements Analysis and Modeling

Richard Y. Wang Henry B. Kon Stuart E. Madnick
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, Mass 02139
rwang@eagle.mit.edu

ABSTRACT

Data engineering is the modeling and structuring of data in its design, development and use. An ultimate goal of data engineering is to put quality data in the hands of users. Specifying and ensuring the quality of data, however, is an area in data engineering that has received little attention. In this paper we: (1) establish a set of premises, terms, and definitions for data quality management, and (2) develop a step-by-step methodology for defining and documenting data quality parameters important to users. These quality parameters are used to determine quality indicators, to be tagged to data items, about the data manufacturing process such as data source, creation time, and collection method. Given such tags, and the ability to query over them, users can filter out data having undesirable characteristics.

The methodology developed provides a concrete approach to data quality requirements collection and documentation. It demonstrates that data quality can be an integral part of the database design process. The paper also provides a perspective for the migration towards quality management of data in a database environment.

1. INTRODUCTION

As data processing has shifted from a role of operations support to becoming a major operation in itself, the need arises for quality management of data. Many similarities exist between quality data manufacturing and quality product manufacturing, such as conformity to specification, lowered defect rates and improved customer satisfaction. Issues of quality product manufacturing have been a major concern for many years [8][20]. Product quality is managed through quality measurements, reliability engineering, and statistical quality control [6][11].

1.1. Related work in data quality management

Work on data quality management has been reported in the areas of accounting, data resource management, record linking methodologies, statistics, and large scale survey techniques. The accounting area focuses on the auditing aspect [3][16]. Data resource management focuses primarily on managing corporate data as an asset [1][12]. Record linking methodologies can be traced to the late 1950ís [18], and have focused on matching records in different files where primary identifiers may not match for the same individual [10][18]. Articles in large scale surveys have focused on data collection and statistical analysis techniques [15][29].

Though database work has not traditionally focused on data quality management itself, many of the tools developed have relevance for managing data quality. For example, research has been conducted on how to prevent data inconsistencies (integrity constraints and normalization theory) and how to prevent data corruption (transaction management) [4][5][9][21]. While progress in these areas is significant, real-world data is imperfect. Though we have gigabit networks, not all information is timely. Though edit checks can increase the validity of data, data is not always valid. Though we try to start with high quality data, the source may only be able to provide estimates with varying degrees of accuracy (e.g., sales forecasts).

In general, data may be of poor quality because it does not reflect real world conditions, or because it is not easily used and understood by the data user. The cost of poor data quality must be measured in terms of user requirements [13]. Even accurate data, if not interpretable and accessible by the user, is of little value.

1.2. A data quality example

Suppose that a sales manager uses a database on corporate customers, including their name, address, and number of employees. An example for this is shown in Table 1.

Co_name	address	#employees
Fruit Co	12 Jay St	4,004
Nut Co	62 Lois Av	700

Table 1: Customer information

Such data may have been originally collected over a period of time, by a variety of company departments. The data may have been generated in different ways for different reasons. As the size of the database grows to hundreds or thousands of records from increasingly disparate sources, knowledge of data quality dimensions such as accuracy, timeliness, and completeness may be unknown. The manager may want to know when the data was created, where it came from, how and why it was originally obtained, and by what means it was recorded into the database. The circumstances surrounding the collection and processing of the data are often missing, making the data difficult to use unless the user of the data understands these hidden or implicit data characteristics.

Towards the goal of incorporating data quality characteristics into the database, we illustrate in Table 2 an approach in which the data is *tagged* with relevant *indicators* of data quality. These quality indicators may help the manager assess or gain confidence in the data.

Co_name	address	#employees
Fruit Co	12 Jay St •1-2-91, sales®	4,004 •10-3-91, Nexis®
Nut Co	62 Lois Av •10-24-91, acctíg®	700 •10-9-91, estimate®

Table 2: Customer information with quality tags

For example, 62 Lois Av, •10-24-91, acctíg® in Column 2 of Table 2 indicates that on October 24, 1991 the accounting department recorded that Nut Coís address was 62 Lois Av. Using such cell-level tags on the data, the manager can make a judgment as to the credibility or usefulness of the data.

We develop in this paper a requirements analysis methodology to both specify the tags needed by users to estimate, determine, or enhance data quality, and to elicit, from the user, more general data quality issues not amenable to tagging. Quality issues not amenable to tagging include, for example, data *completeness* and *retrieval time*. Though not addressable via cell-level tags, knowledge of such dimensions can aid data quality control and systems design. (Tagging higher aggregations, such as the table or database level, may handle some of these more general quality concepts. For example, the means by which a database table was populated may give some indication of its completeness.)

Formal models for cell-level tagging, the attribute-based model [28] and the polygen source-tagging model [24][25], have been developed elsewhere. The function of these models is the tracking of the production history of the data artifact (i.e., the processed electronic symbol) via tags. These models include data structures, query processing, and model integrity considerations. Their approach demonstrates that the data manufacturing process can be modeled independently of the application domain.

We develop in this paper a methodology to determine which aspects of data quality are important, and thus what kind of tags to put on the data so that, at query time, data with undesirable characteristics can be filtered out. More general data quality issues such as data quality assessment and control are beyond the scope of the paper.

The terminology used in this paper is described next.

1.3. Data quality concepts and terminology

Before one can analyze or manage data quality, one must understand what data quality means. This can not be done out of context, however. Just as it would be difficult to manage the quality of a production line without understanding dimensions of product quality, data quality management requires understanding which dimensions of data quality are important to the user.

It is widely accepted that quality can be defined as ìconformance to requirementsî [7]. Thus, we define data quality on this basis. Operationally, we define data quality in terms of data quality parameters and data quality indicators (defined below).

ï A **data quality parameter** is a qualitative or subjective dimension by which a user evaluates data quality. *Source credibility* and *timeliness* are examples. (called *quality parameter* hereafter)

ï A **data quality indicator** is a data dimension that provides objective information about the data. *Source*, *creation time,* and *collection method* are examples. (called *quality indicator* hereafter)

ï A **data quality attribute** is a collective term including both quality parameters and quality indicators, as shown in Figure 1 below. (called *quality attribute* hereafter)

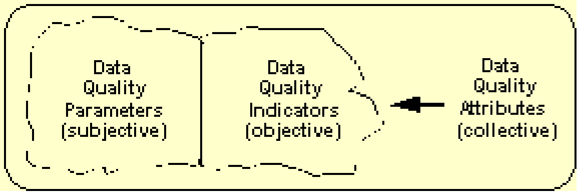
Data Quality Parameters (subjective)
Data Quality Indicators (objective)
Data Quality Attributes (collective)

Figure 1: Relationship among quality attributes, parameters, and indicators

ï A **data quality indicator value** is a measured characteristic of the stored data. The data quality indicator *source* may have an indicator value *Wall Street Journal*. (called *quality indicator value* hereafter)

ï A **data quality parameter value** is the value determined for a quality parameter (directly or indirectly) based on underlying quality indicator values. User-defined functions may be used to map quality indicator values to quality parameter values. For example, because the *source* is *Wall Street Journal*, an investor may conclude that data *credibility* is *high*. (called *quality parameter value* hereafter)

ï **Data quality requirements** specify the indicators required to be tagged, or otherwise documented for the data, so that at query time users can retrieve data of specific quality (i.e., within some acceptable range of quality indicator values). (called *quality requirements* hereafter)

ï The **data quality administrator** is a person (or system) whose responsibility it is to ensure that data in the database conform to the quality requirements.

For brevity, the term "quality" will be used to refer to "data quality" throughout this paper.

2. FROM DATA MODELING TO DATA QUALITY MODELING

It is recognized in manufacturing that the earlier quality is considered in the production cycle, the less costly in the long run because upstream defects cause downstream inspection, rework, and rejects [22]. The lesson to data engineering is to design data quality into the database, i.e., quality data by design.

In traditional database design, aspects of data quality are not explicitly incorporated. Conceptual design focuses on application issues such as entities and relations. As data increasingly outlives the application for which it was initially designed, is processed along with other data, and is used over time by users unfamiliar with the data, more explicit attention must be given to data quality. Next, we present premises related to data quality modeling.

In general, different users have different data quality requirements, and different data is of different quality. We present related premises in the following sections.

2.1. Premises related to data quality modeling

Data quality modeling is an extension of traditional data modeling methodologies. Where data modeling captures the structure and semantics of data, data quality modeling captures structural and semantic issues underlying data quality.

•**Premise 1.1® Relatedness of application and quality attributes:** Application attributes and quality attributes may not always be distinct. For example, the name of the bank teller who performs a transaction may be considered an application attribute. Alternatively, it may be modeled as a quality indicator to be used for data quality administration. Thus, we identify two distinct domains of activity: data usage and quality administration. If the information relates to aspects of the data manufacturing process, such as when, where, and by whom the data was manufactured, then it may be a quality indicator.

•**Premise 1.2® Quality attribute non-orthogonality:** Different quality attributes need not be orthogonal to one another. For example, the two quality parameters *timeliness* and *volatility* are related.

•**Premise 1.3® Heterogeneity and hierarchy in the quality of supplied data:** Quality of data may differ across databases, entities, attributes, and instances. Database example: data in the *alumni* database may be less *timely* than data in the *student* database. Attribute example: in the student entity, *grades* may be more *accurate* than *addresses*. Instance example: data about an *international student* may be less *interpretable* than that of a *domestic student*.

•**Premise 1.4® Recursive quality indicators:** One may ask ìwhat is the quality of the quality indicator values?î In this paper, we ignore the recursive notion of *meta-quality* *indicators*, as our main objective is to develop a quality perspective in requirements analysis. This is a valid issue, however, and is handled in [28] where the same tagging and query mechanism applied to application data is applied to quality indicators.

2.2. Premises related to data quality definitions and standards across users

Because human insight is needed for data quality modeling and because people have individual opinions about data quality, different quality definitions and standards exist across users. The users of a given (local) system may know the quality of the data they use. When data is exported to other users, however, or combined with information of different quality, data quality may become unknown, leading to different needs in quality attributes across application domains and users. The following two premises discuss that ìdata quality is in the eye of the beholder.î

•**Premise 2.1® User specificity of quality attributes:** Quality parameters and quality indicators may vary from one user to another. Quality parameter example: for a *manager* the critical quality parameter for a *research report* may be *cost*, whereas for a *financial trader, credibility* and *timeliness* may be more critical. Quality indicator example: the manager may measure *cost* in terms of the quality indicator *(monetary) price*, whereas the trader may measure cost in terms of opportunity cost or competitive value of the information, and thus the quality indicator may be *age* of the data.

•**Premise 2.2® Users have different quality standards:** Acceptable levels of data quality may differ from one user to another. An investor loosely following a stock may consider a ten minute delay for *share price* sufficiently timely, whereas a trader who needs price quotes in real time may not consider ten minutes timely enough.

2.3. Premises related to a single user

Where Premises 2.1 and 2.2 stated that different users may specify different quality attributes and standards, a single user may specify different quality attributes and standards for different data. This is summarized in Premise 3 below.

•**Premise 3® For a single user; non-uniform data quality attributes and standards:** A user may have different quality attributes and quality standards across databases, entities, attributes, or instances. Across attributes example: a user may need higher quality information for *address* than for the *number of employees.* Across instances example: an analyst may need higher quality information for certain companies than for others as some companies may be of particular interest.

3. DATA QUALITY MODELING

We now present the steps in data quality modeling. In Section 2, we described data quality modeling as an effort similar in spirit to traditional data modeling, but focusing on quality aspects of the data. As a result of this similarity, we can draw parallels between the database life cycle [23] and the requirements analysis methodology developed in this paper.

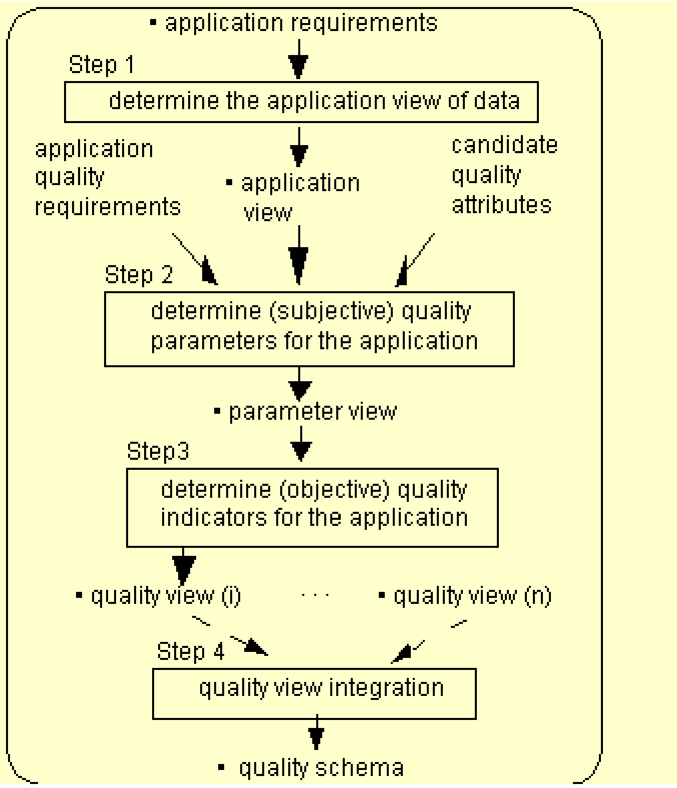
▪ application requirements
Step 1
determine the application view of data
application quality requirements
▪ application view
candidate quality attributes
Step 2
determine (subjective) quality parameters for the application
▪ parameter view
Step3
determine (objective) quality indicators for the application
▪ quality view (i)
. . .
▪ quality view (n)
Step 4
quality view integration
▪ quality schema

Figure 2: The process of data quality modeling

The final outcome of data quality modeling, the *quality schema*, documents both application data requirements and data quality issues considered important by the design team. The methodology guides the design team as to which tags to incorporate into the database. Determination of acceptable quality levels (i.e., filtering of data by quality indicator values) is done at query time. Thus, the methodology does not require the design team to define cut-off points, or acceptability criteria by which data will be filtered. The overall methodology is diagrammed above in Figure 2. For each step, the input, output and process are included.

A detailed discussion of each step is presented in the following sections.

3.1. Step 1: Establishing the application view

Input: application requirements

Output: application view

Process: This initial step embodies the traditional data modeling process and will not be elaborated upon here. A comprehensive treatment of the subject has been presented elsewhere [17][23]. The objective is to elicit and document application requirements of the database.

We will use the following example application throughout this section (Figure 3). Suppose a stock trader keeps information about companies, and *trades* of *company stocks* by *clients*. Client is identified by an *account number*, and has a *name*, *address*, and *telephone number*. Company stock is identified by the companyís *ticker symbol*, and has *share price* and *research report* associated with it. When a client makes a *trade* (*buy/sell*), information on the *date*, *quantity* of shares and *trade price* is stored as a record of the transaction. The ER application view for the example application is shown in Figure 3 below.

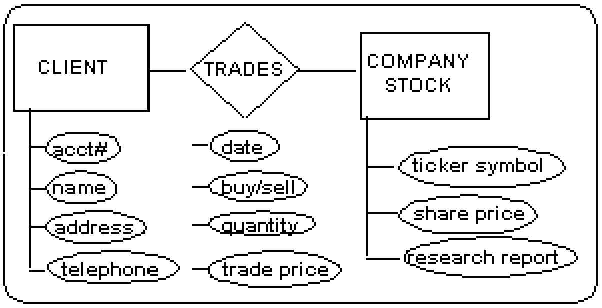

Figure 3: Application view (output from Step 1)

3.2. Step 2: Determine (subjective) quality parameters

Input: application view, application quality requirements, candidate quality attributes

Output: parameter view (quality parameters added to the application view)

Process: The goal here is to elicit data quality needs, given an application view. For each component of the application view, the design team should determine those quality parameters needed to support data quality requirements. For example, *timeliness* and *credibility* may be two important quality parameters for data in a trading application.

Appendix A provides a list of candidate quality attributes for consideration in this step. The list resulted from survey responses from several hundred data users asked to identify facets of the term ìdata qualityî [26]. Though items in the list are not orthogonal, and the list is not provably exhaustive, the aim here is to stimulate thinking by the design team about data quality requirements. Data quality issues relevant for future and alternative applications should also be considered at this stage. The design team may choose to consider additional parameters not listed.

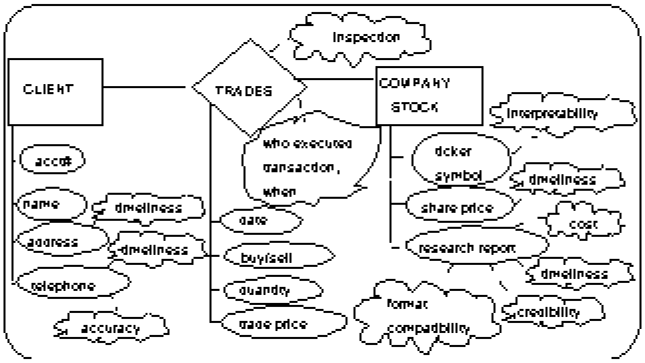

Figure 4: Parameter view: quality parameters added to application view (output from Step 2)

An example parameter view for the application is shown above in Figure 4. Each parameter is inside a ìcloudî in the diagram. For example, t*imeliness* on *share price* indicates that the user is concerned with how old the data is; *cost* for the *research report* suggests that the user is concerned with the price of the data. A special symbol, "_ inspection" is used to signify inspection (e.g., data verification) requirements.

Quality parameters identified in this step are added to the application view resulting in the *parameter view.* The parameter view should be included as part of the quality requirements specification documentation.

3.3. Step 3: Determine (objective) quality indicators

Input: parameter view (the application view with quality parameters included)

Output: quality view (the application view with quality indicators included)

Process: The goal here is to operationalize the subjective quality parameters into measurable or precise characteristics for tagging. These measurable characteristics are the quality indicators. Each quality indicator is depicted as a dotted-rectangle (Figure 5) and is linked to the entity, attribute, or relation where there was previously a quality parameter.

It is possible that during Step 2, the design team may have defined some quality parameters that are somewhat objective. If a quality parameter is deemed in this step to be sufficiently objective (i.e., can be directly operationalized), it can remain. For example, if *age* had been defined as a quality parameter, and is deemed objective, it can remain as a quality indicator. Quality indicators replace the quality parameters in the parameter view, creating the *quality view*.

From Figures 4 and 5; corresponding to the quality parameter *timeliness*, is the more objective quality indicator *age* (of the data). The *credibility* of the research report is indicated by the quality indicator *analyst name.*

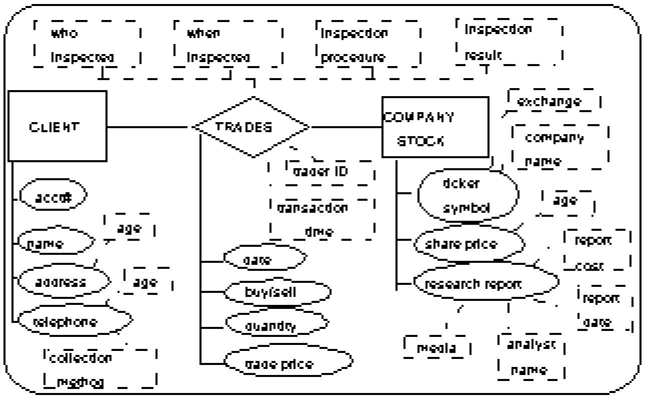

Figure 5: Quality View: quality indicators added to application view (output from Step 3)

Note the quality indicator *collection method* associated with the *telephone* attribute. It is included to illustrate that multiple data collection mechanisms can be used for a given type of data. In the telephone example, values for collection method may include ìover the phoneî or ìfrom an information serviceî. In general, different means of capturing data such as bar code scanners in supermarkets, radio frequency readers in the transportation industry, and voice decoders each has inherent accuracy implications. Error rates may differ from device to device or in different environments. The quality indicator *media* for *research report* is to indicate the multiple formats of database-stored documents such as bit mapped, ASCII or post script.

The quality indicators derived from the "_ inspection" quality parameter indicate the inspection mechanism desired to maintain data reliability. The specific inspection or control procedures may be identified as part of the application documentation. These procedures might include double entry of important data, front-end rules to enforce domain or update constraints, or manual processes for performing certification on the data.

The resulting quality view, together with the parameter view, should be included as part of the quality requirements specification documentation.

3.4. Step 4: Perform quality view integration (and application view refinement)

Input: quality view(s)

Output: (integrated) quality schema

Process: Much like schema integration [2], when the design is large and more than one set of application requirements is involved, multiple quality views may result. To eliminate redundancy and inconsistency, these views must be consolidated into a single global view so that a variety of data quality requirements can be met.

This involves the integration of quality indicators. In simpler cases, a union of these indicators may suffice. In more complicated cases, such as non-orthogonal quality attributes, the design team may examine the relationships among the indicators in order to decide what kind of indicators to be included in the integrated quality schema. For example, one quality view may have *age* as an indicator, whereas another quality view may have *creation time*. In this case, the design team may choose creation time for the integrated schema because age can be computed given current time and creation time.

Another task that needs to be performed at this stage is a re-examination of the structural aspect of the schemas (Premise 1.1). In the example application, for instance, *company name* is not specified as an entity attribute of *company stock* (in the application view) but rather appears as a quality indicator to enhance the interpretability of *ticker symbol*. After re-examining the application requirements, the design team may conclude that company name should be included as an entity attribute of company stock instead of a quality indicator for *ticker symbol*.

In the example application, because only one set of requirements is considered, only one quality view results and there is no view integration. The resulting integrated quality schema, together with the component quality views and parameter views, should be included as part of the quality requirements specification documentation.

This concludes the four steps of the methodology for data quality requirements analysis and modeling.

4. DISCUSSION

The data quality modeling approach developed in this paper provides a foundation for the development of a quality perspective in database design. End-users need to extract quality data from the database. The data quality administrator needs to monitor, control, or report on the quality of information.

Users may choose to only retrieve or process information of a specific ìgradeî (e.g., provided recently via a reliable collection mechanism) or inspect data quality indicators to determine how to interpret data [28]. Data quality profiles may be stored for different applications.

For example, an information clearing house for addresses of individuals may have several classes of data. For a *mass mailing* application there may be no need to reach the correct individual (by name), and thus a query with no constraints over quality indicators may be appropriate. For more sensitive applications, such as *fund raising*, the user may query over and constrain quality indicators values, raising the accuracy and timeliness of the retrieved data.

The administratorís perspective is in the area of inspection and control. In handling an exceptional situation, such as tracking an erred transaction, the administrator may want to track aspects of the data manufacturing process, such as the time of entry or intermediate processing steps. Much like the ìpaper trailî currently used in auditing procedures, an ìelectronic trailî may facilitate the auditing process. The ìinspectionî indicator is intended to encompass issues related to the data quality management function. Specifications may be included such as those for statistical process control, data inspection and certification, data-entry controls, and potentially include process-based mechanisms such as prompting for data inspection on a periodic basis or in the event of peculiar data.

Developing a generalizable definition for dimensions of data quality is desirable. Certain characteristics seem universally important such as completeness, timeliness, accuracy, and interpretability. Some of the items listed in Appendix A, however, apply more to the information system (resolution of graphics), the information service (clear data responsibility), or the information user (past experience), than to the data itself. Where one places the boundary of the concept of data quality will determine which characteristics are applicable. The derivation and estimation of quality parameter values and overall data quality from underlying indicator values remains an area for further investigation.

Organizational and managerial issues in data quality control involve the measurement or assessment of data quality, analysis of impacts on the organization, and improvement of data quality through process and systems redesign and organizational commitment to data quality [13][27]. Cost-benefit tradeoffs in tagging and tracking data quality must be considered. Converging on standardized data quality attributes may be necessary for data quality management in cases where data is transported across organizations and application domains.

These additional implementation and organizational issues are critical to the development of a quality control perspective in data processing.

5. CONCLUSION

In this paper we have established a set of premises, terms, and definitions for data quality, and developed a step-by-step methodology for data quality requirements analysis, resulting in an ER-based quality schema. This paper contributes in three areas. First, it provides a methodology for data quality requirements collection and documentation. Second, it demonstrates that data quality can be included as an integral part of the database design process. Third, it offers a perspective for the migration from todayís focus on the application domain towards a broader concern for data quality management.

6. REFERENCES

[1] Ballou, D. P. & Kumar.Tayi, G. (1989). Methodology for Allocating Resources for Data Quality Enhancement. Communications of the ACM, 32(3), pp. 320-329.

[2] Batini, C., Lenzirini, M., & Navathe, S. (1986). A comparative analysis of methodologies for database schema integration. ACM Computing Surveys, 18(4), pp. 323-364.

[3] Bodner, G. (1975). Reliability Modeling of Internal Control Systems. Accounting Review, 50(October), pp. 747-757.

[4] Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), pp. 377-387.

[5] Codd, E. F. (1986). An evaluation scheme for database management systems that are claimed to be relational. The Second International Conference on Data Engineering, Los Angeles, CA. 1986. pp. 720-729.

[6] Crosby, P. B. (1979). Quality is Free. New York: McGraw-Hill.

[7] Crosby, P. B. (1984). Quality Without Tears. New York: McGraw-Hill Book Company.

[8] Deming, W. E. (1982). Quality, Productivity, and Competitive Position. Cambridge, MA: MIT Center for Advanced Engineering Study.

[9] Elmasri, R. (1989). Fundamentals of Database Systems. Reading, MA: The Benjamin/Cummings Publishing Co., Inc.

[10] Fellegi, I. P. & Sunter, A. B. (1969). A Theory for Record Linkage. Journal of the American Statistical Association, 64(number 328), pp. 1183-1210.

[11] Garvin, D. A. (1988). Managing Quality-The Strategic and Competitive Edge (1 ed.). New York: The Free Press.

[12] Goodhue, D. L., Quillard, J. A., & Rockart, J. F. (1988). Managing The Data Resources: A Contingency Perspective. MIS Quarterly, 12(3), pp. 373-392.

[13] Ishikawa, K. (1985). What is Total Quality Control?-the Japanese Way. Englewood Cliffs, NJ. Prentice-Hall

[14] Laudon, K. C. (1986). Data Quality and Due Process in Large Interorganizational Record Systems. Communications of the ACM, 29(1), pp. 4-11.

[15] Liepins, G. E. & Uppuluri, V. R. R. (1990). Data Quality Control: Theory and Pragmatics. New York: Marcel Dekker, Inc.

[16] Lindgren, B. (1991). Getting data with integrity. Enterprise, (Winter), pp. 30-34.

[17] Navathe, S., Batini, C., & Ceri, S. (1992). The Entity Relationship Approach. New York: Wiley and Sons.

[18] Newcombe, H. B. & Kennedy, J. M. (1962). Record Linkage: Making maximum use of the discriminating power of identifying information. Communications of the ACM, 5.

[19] Newcombe, H. B., et al. (1959). Automatic Linkage of Vital Records. Science, 130(3381), pp. 954-959.

[20] Shewhart, W. A. (1925). The Application of Statistics as an Aid in Maintaining Quality of a Manufactured Product. Journal of the American Statistical Association, 20, pp. 546-548.

[21] Stonebraker, M. & Kemnitz, G. (1991). The Postgres Next-Generation Database Management System. Communications of the ACM, 34(10), pp. 78-92 (October).

[22] Taguchi, G. (1979). Introduction to Off-line Quality Control. Magaya, Japan: Central Japan Quality Control Association.

[23] Teorey, T. J. (1990). Database Modeling and Design: The Entity-Relationship Approach. San Mateo, CA : Morgan Kaufman Publisher.

[24] Wang, Y. R. & Madnick, S. E. (1990). A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective, 16th Conference on Very Large Data Bases, Brisbane, Australia. 1990. pp. 519-538.

[25] Wang, Y. R. & Madnick, S. E. (1990). A Source Tagging Theory for Heterogeneous Database Systems, International Conference on Information Systems, Copenhagen, Denmark. 1990. pp. 243-256.

[26] Wang, Y. R. & Guarrascio, L. M. (1991). Dimensions of Data Quality: Beyond Accuracy. (CISL-91-06) Composite Information Systems Laboratory, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, 02139 June 1991.

[27] Wang, R. Y. & Kon, H. B. (1992). Towards Total Data Quality Management (TDQM). In R. Y. Wang (Ed.), Englewood Cliffs, NJ: Prentice Hall.

[28] Wang, Y. R. & Reddy, M. P., Kon, H.B. (1993). Toward Quality Data: An Attribute-based Approach, to appear in the Journal of Decision Support Systems (DSS), Special Issue on Information Technologies and Systems

[29] Zarkovich. (1966). Quality of Statistical Data. Rome: Food and Agriculture Organization of the United Nations.

Appendix A: Candidate Data Quality Attributes (input to Step 2)

Ability to be Joined With Ability to Download Ability to Identify Errors Ability to Upload

Acceptability Access by Competition Accessibility Accuracy

Adaptability Adequate Detail Adequate Volume Aestheticism

Age Aggregatability Alterability Amount of Data

Auditable Authority Availability Believability

Breadth of Data Brevity Certified Data Clarity

Clarity of Origin Clear Data Responsibility Compactness Compatibility

Competitive Edge Completeness Comprehensiveness Compressibility

Concise Conciseness Confidentiality Conformity

Consistency Content Context Continuity

Convenience Correctness Corruption Cost

Cost of Accuracy Cost of Collection Creativity Critical

Current Customizability Data Hierarchy Data Improves Efficiency

Data Overload Definability Dependability Depth of Data

Detail Detailed Source Dispersed Distinguishable Updated Files

Dynamic Ease of Access Ease of Comparison Ease of Correlation

Ease of Data Exchange Ease of Maintenance Ease of Retrieval Ease of Understanding

Ease of Update Ease of Use Easy to Change Easy to Question

Efficiency Endurance Enlightening Ergonomic

Error-Free Expandability Expense Extendibility

Extensibility Extent Finalization Flawlessness

Flexibility Form of Presentation Format Format Integrity

Friendliness Generality Habit Historical Compatibility

Importance Inconsistencies Integration Integrity

Interactive Interesting Level of Abstraction Level of Standardization

Localized Logically Connected Manageability Manipulable

Measurable Medium Meets Requirements Minimality

Modularity Narrowly Defined No lost information Normality

Novelty Objectivity Optimality Orderliness

Origin Parsimony Partitionability Past Experience

Pedigree Personalized Pertinent Portability

Preciseness Precision Proprietary Nature Purpose

Quantity Rationality Redundancy Regularity of Format

Relevance Reliability Repetitive Reproducibility

Reputation Resolution of Graphics Responsibility Retrievability

Revealing Reviewability Rigidity Robustness

Scope of Info Secrecy Security Self-Correcting

Semantic Interpretation Semantics Size Source

Specificity Speed Stability Storage

Synchronization Time-independence Timeliness Traceable

Translatable Transportability Unambiguity Unbiased

Understandable Uniqueness Unorganized Up-to-Date

Usable Usefulness User Friendly Valid

Value Variability Variety Verifiable

Volatility Well-Documented Well-Presented