DATABASE QUALITY DIMENSIONS

John Hoxmier*

To ensure a quality database application, should the emphasis during model development be on the application of quality assurance metrics (designing it right)? It's hard to argue against this point, but there is a significant amount of research and anecdotal evidence that suggests that a large number of organizational database applications fail or are unusable. A quality process does not necessarily lead to a usable database product. Databases are a critical element of virtually all conventional and ebusiness applications A database should be evaluated in production based on certain quantitative and information-preserving transformation measures, such as data quality, data integrity, normalization, and performance. However, there are also many examples of database applications that are in most ways 'well-formed' with high data quality but lack semantic or cognitive fidelity (the right design). Additionally, determining and implementing the proper set of database behaviors can be an elusive task. Whether the database meets the expectations of its end-users is only one aspect of overall database quality. This paper expands on the growing body of literature in the area of data quality by proposing additions to a hierarchy of database quality dimensions that includes model and behavioral factors in addition to the process and data factors.

Most information systems depend on a database to record and retrieve application data and preserve organizational memory. The ultimate objective of database analysis, design, and implementation is to establish an electronic repository that is a physical and behavioral model of the manageable aspects of a user's information domain. Database design is a complex, complicated art. Many factors must be considered during the process including, but not limited to, historical and future information requirements, the diversity of the data consumer community, organizational requirements, security, cost, ownership, performance, interface issues, and data integrity. These factors contribute to the success of a database application in both quantitative and qualitative ways and determine the

* *John Hoxmier is affiliated with Colorado State University.*

overall quality of the database application. *Process* and *data* quality is quantitative management factors that are fairly well documented and understood, albeit underutilized. However, data *model* and *behavioral* considerations include important qualitative factors that contribute to overall database quality. A database is more than the instances of the data it manages. Data quality, while important, is just one element of assessing overall database quality.

This paper expands on the growing body of literature in the area of data quality by proposing additions to a hierarchy of database quality dimensions that includes model and behavioral factors in addition to the process and data factors. The term "database quality" in this context expands on the ISO definition of quality, i.e. *conformance to requirements* and *fitness for use* (1993). This definition is not adequate for the purposes of assessing database quality. While the requirement definition phase of the system development life cycle is critical to the success of an application, doing a good job of defining requirements is not sufficient in the implementation of a successful database application. A database must also be judged by how closely it represents the world of the data consumer (the model) and its ability to respond to both routine and unanticipated requests within the domain it is expected to manage (the behavior). The framework presented herein expands on work previously proposed (Hoxmeier, 1997; Hoxmeier and Monarchi, 1996) and incorporates data quality dimensions put forth by several prominent data quality researchers (Ballou and Pazar, 1995; Storey and Wang, 1994; Strong, et al., 1997; Wand and Wang, 1996; Wang et al., 1993; Wang, et al., 1995).

The Problem/Solution Cycle

The database design process is largely driven by the requirements and needs of the consumer, who establishes the boundaries and properties of the problem domain and the requirements of the information. As organizations seek to preserve organizational memory and manage richer forms of information over broader networks, this task has become increasingly more difficult.

Figure 1. Problem to Solution Cycle

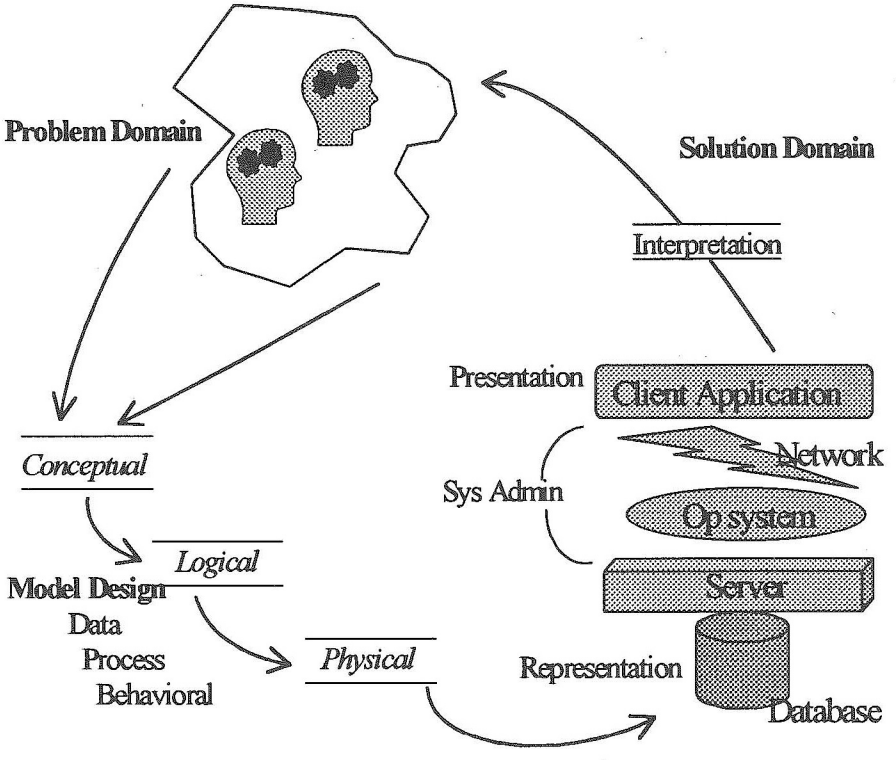


Figure 1 illustrates a typical scenario in the problem to solution cycle. It is not difficult to see why so many database applications are ultimately unsuitable to the consumer. Designers attempt to conceptualize the problem domain into a suitable physical implementation. The proposed solution is subject to many constraints including the physical representation, system administration, application presentation, and information interpretation. These constraints or solution layers all contribute to the perceived quality of the solution by the information consumer. Figure 1 also shows the critical elements in the problem to solution cycle that are the bases for the discussion on database quality dimensions:

- The cycle *process* must be managed toward a successful outcome.
- The *model* itself must represent a usually diverse and fuzzy problem domain.
- The quality of the *data* in the database must be of sufficient grade.
- The application must *behave* in a way the consumer understands.

The last step depicted in the illustration, interpretation, is probably outside of the direct control of the design and development team. However, the consumer's ability to interpret the information is also critical to the success of a database application and, therefore, to the perceived quality of the database.

To ensure a quality database application, should the emphasis during model development be on the application of quality assurance metrics (designing it right)? It's hard to argue against this point, but there are a significant number of studies and anecdotal evidence that suggests that a large number of database applications fail or are unusable (Standish Group, 1997; Wand and Wang, 1996). A quality process does not necessarily lead to a usable database product (Hoxmeier, 1995; Redman, 1995). A database should be evaluated in production based on certain quantitative and information-preserving transformation measures, such as data quality, data integrity, normalization, and performance. However, there are also many examples of database applications that are in most ways 'well-formed' with high data quality but lack semantic or cognitive fidelity (the right design). Additionally, determining and implementing the proper set of database behaviors can be an elusive task. Depending on the risk factors affecting the application, there may be certain aspects of the quality assessment that deserve heavier weights. Contrary to the popular notion of product quality, whether the database meets the expectations of its end-users is only one aspect of overall database quality.

Significant Prior Research

Quality metrics have been used for years in the design, development, and marketing for consumer goods and services. Quality engineering methods, such as Total Quality Management (TQM) and Quality Function Deployment (QFD) are commonly used by many product design and manufacturing disciplines, and are rapidly entering the service disciplines. In the area of information quality, however, the use of these techniques is virtually non-existent. Recently, researchers have begun to evaluate and study the characteristics of information as they would any other product or service (Wang et al., 1995).

Researchers and practitioners alike have tried to establish a set of factors, attributes, rules or guidelines in order to evaluate system quality. Zmud concluded that a set of four dimensions divided into 25 factors represented the dimensions of information quality (Zmud, 1978). The dimensions included data quality, relevancy, format quality, and meaning quality. Referring to information systems, James Martin stated that the collection of data has little value unless the data are used to understand the world and prescribe action to improve it (Martin, 1976).

Cap Gemini Pandata, a Dutch company, uses a framework that decomposes the entire information quality notion into four dimensions, 21 aspects, and 40 attributes (Delen and Rijsenbrij, 1992). Cap Gemini has adopted this framework on the company procedures covering software package auditing. AT&T is researching data quality and have identified four primary factors including accuracy, currentness, completeness and consistency (Fox, et al., 1994). Another group, the Southern California Online Users Group (SCOUG), defined characteristics of a quality library online database (Tenopir, 1990). The purpose of the set of characteristics was to allow professional searchers to rate each library online database system.

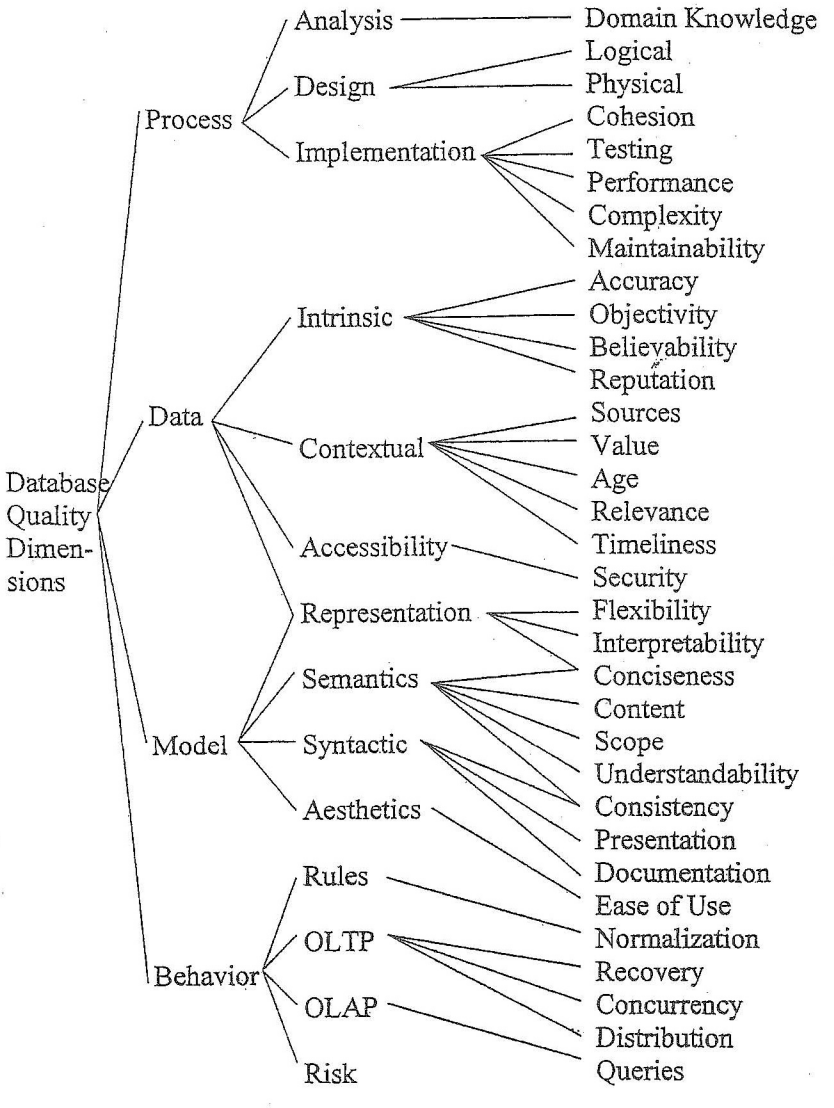
Marketing research has identified approaches used to assess product quality attributes that are important to consumers (Churchill, 1991; Menon, 1997). Wang et al. applied this concept toward a data consumer (1996). They performed a comprehensive survey that identified 4 high-level categories of data quality after evaluating 118 variables). The Wang factors include intrinsic data quality, contextual data quality, representation data quality, and accessibility data quality. A recent study applied the model to a series of field studies that focused on the concerns of the data consumer (Strong et al., 1997). These field studies confirmed the dimensions of data quality set forth in the Wang study.

There appear to be many similarities in the factors identified in these studies based on the perspective of the evaluators. Both developers and data consumers are concerned with data quality metrics like accuracy, timeliness, and consistency. Most of the research, while focused on data or information quality, indicates that there are a diverse set of factors influencing data quality. Any individual variable however, such as accuracy, is difficult to quantify. Nonetheless, researchers have developed a fairly consistent view of data quality. There is little available in the literature on the evaluation of overall database quality including other considerations such as semantic fidelity, behavioral, and value factors.

The Proposed Framework

It is proposed that through the hierarchical framework presented below, one can evaluate overall database quality by assessing four primary dimensions: process, data, model, and behavior. Portions of the hierarchy draw heavily from previous studies on data and information quality, and documented process quality standards. A dimension is a set of database quality attributes or components that most data consumers react to in a fairly consistent way (Wang, et al., 1996). The use of a set of dimensions to represent a quality typology is consistent with previous quality research (Dvir and Evans, 1996; Wang, et al., 1996; Strong, et al., 1997). The framework presents the four dimensions in a dimension-attribute-property hierarchy.

Figure 2. Database Quality Dimensions



Process Quality

Much attention has been given over the years to process quality improvement. *ISO-9000-3, Total Quality Management (TQM), and Quality Function Deployment (QFD)* are approaches that are concerned primarily with the incorporation of quality management within the process of systems development (Costin, 1994; Dvir and Evans, 1996; Schmauch, 1994). *Quality control* is a process of ensuring that the database conforms to predefined standards and guidelines using statistical quality measures (Dyer, 1992). It compares variations of identified activities with the results of predetermined standards and assesses the variation between the two. When deviations from the problem domain are found, they are resolved and the process is modified as needed. This is an effective, yet reactive form of quality management. *Quality assurance* attempts to maintain the quality standards in a proactive way. In addition to using quality control measures, quality assurance goals go further by surveying the customer to determine their level of satisfaction with the product. Conceivably, potential problems can be detected early in the process.

The philosophy of ISO-9000-3 is to build quality into a software system on a continuous basis, from conception through implementation. ISO-9000-3 as a process quality standard does not offer any particular metrics to be utilized during the process. In addition, as a general software standard, ISO-9000-3 does not deal specifically with database issues.

A specific property addition to the framework within the dimension of process implementation quality is performance. All too often, specific performance requirements are either ignored during the design process or evaluated after implementation. While performance, per se, is more of an implementation issue, it should be considered as an aspect of overall database quality, even in the conceptual phase. Both relational and object databases can contain rather serious problems in terms of data redundancy, relationships, integrity, and structure. The objective is to design a normalized, high-fidelity database while minimizing complexity. When evaluating performance there are times when de-normalization may represent an optimal solution. However, anytime a general-purpose database is optimized for a given situation, other requirements inevitably arise that negate the advantage. The measures used to assess the trade-off may include query and update performance, storage, and the avoidance of data anomalies. Similar to the contrast between data and semantic quality, a database that is otherwise well designed but does not perform well is useless.

Database Data Quality

Data integrity is one of the keys to developing a quality database. Without accurate data, users will lose confidence in the database or make uninformed decisions (Redman, 1995). While data integrity can become a problem over time, there are relatively straightforward ways to enforce constraints and domains and to ascertain when problems exist (Moriarty, 1996). The identification, interpretation, and application of business rules, however, present a more difficult challenge for the developer. Rules and policies must be communicated and translated and much of the meaning and intent can be lost in this process. Because data quality has been a focus of previous research (for an excellent discussion, see Strong et al., 1997) and these studies have been used as a basis for the data dimension presented here, the individual attributes will not be discussed. However, a couple of additional properties are worth noting.

A frequently overlooked metric in the evaluation of data integrity is the age of the data, database, and model. Data or model age is different than the timeliness property. Timeliness refers to the delay between availability and accessibility. Age refers to the time that has passed since the data was entered into the database or when the data model was developed. The data should only be as old as the problem domain and information sources will allow and maintained only as long as the situation requires. This can be a few seconds or several years. At some point, the data needs to be refreshed in order to maintain its currency. Over time, the age of the model may degrade in its ability to depict the problem domain. The model must be updated so that as the problem domain changes, the model of the database changes as well.

Additionally, the assessment of data quality must include value considerations. Time and financial constraints are real concerns. As IT departments are expected to do more with less and as cycle times continue to decrease for database applications, developers must make decisions about the extent to which they are going to implement and evaluate quality considerations. Shorter cycle times present a good argument for modularity and reusability, so quality factors must be addressed on a micro basis.

Data Model Quality

As has been presented, data quality is usually associated with the quality of the data values. However, even data that meets all other quality criteria is of little use if it is based on a deficient data model (Levitin and Redman, 1995). Data model quality is the third of the four high level dimensions presented above. Information and an application that represent a high proportionate match between the problem and solution domains should be the goal of a database with high semantic quality. Representation, semantics, syntax, and aesthetics are all attributes of model quality (Hoxmeier and Monarchi, 1996; Levitin and Redman, 1995; Lindland et al., 1994).

The database design process is largely driven by the requirements and needs of the data consumer, who establishes the boundaries and properties of the problem domain and the requirements of the task. The first step in the process, information discovery, is one of the most difficult, important, and labor intensive stages of database development (Chignell and Parsaye, 1993; Sankar and Marshall, 1993). It is in this stage where the semantic requirements are identified, prioritized, and visualized. Requirements can rarely be defined in a serial fashion. Generally, there is significant uncertainty over what these requirements are, and they only become clearer after considerable analysis, discussions with users, and experimentation with prototypes. This means previous work may be revisited. Additionally, while many studies point to the importance of user involvement in the discovery and design phase, many information consumers are uncertain about their requirements or have insufficient database knowledge to provide much insight.

Concentric design is an approach that is appropriate in database design. This cyclical process emulates the philosophy of continuous quality improvement used in Total Quality Management (Braithwaite, 1994; Dvir and Evans, 1994). The costs associated with developing quality into the application from design to implementation are much lower than the costs of correcting problems that occur later due to poor design. However, the learning curve within the domain for the designer may be steep and the demand for the application may force rapid delivery. So, how do designers arrive at high semantic quality in a very short period of time?

Qualitative techniques address the ambiguous and subjective dimensions of conceptual database design. The interaction between people and information is one where human preference and constraints have a huge impact on the effectiveness of database design. The use of techniques such as affinity and pareto diagrams, semantic object models, group decision support systems, nominal group, and interrelationship digraphs help to improve the process of problem and solution domain definition. Well studied quantitative techniques, such as entity-relationship diagrams, object models, data flow diagrams, and performance benchmarks, on the other hand, allow the results of the qualitative techniques to be described in a visual format and measured in a meaningful way. Other object attributes that explicitly express quality can be included in the model as well. Storey and Wang present an innovative extension to the traditional ER approach for incorporating quality requirements (database quality data and product quality data) into conceptual database design (1994). The underlying premise of the approach is that quality requirements should be distinct from other database properties.

These techniques can be used to assist the developer extract a strong semantic model. However, it is difficult to design a database with high semantic value without significant domain knowledge and experience (Navathe, 1997). These may be the two most important considerations in databases of high semantic quality. In addition, conceptual database design remains more of an art than a science. It takes a high amount of creativity and vision to design a solution that is robust, usable, and can stand the test of time.

Database Behavior Quality

Many databases are perceived to be of low quality simply because they are difficult to use. In a recent survey in the UK, managers and professionals from various disciplines were asked to evaluate the quality of information they were using (Rolph and Bartram, 1994). Using eight factors, "accuracy" rated the highest, "usable format" the lowest. Developers tend to focus on aspects of data quality at the expense of behavioral quality. Granted, the behaviors associated with a general-purpose database used for decision and analytical support are varied and complex.

What constitutes a database of high behavioral quality? Are the criteria different than those used for software applications in general? Clearly the behaviors for a database that is used to support transaction processing (OLTP) are different than those of a database used to support analytical processing (OLAP). Software development, in general, is very procedure- or function-driven. The objective is to build a system that works (and do it quickly). Database development, on the other hand, should be more focused on the content, context, behavior, semantics, and persistence of the data. Rapid application development and prototyping techniques contribute to arriving at a close match between the problem and solution domains. There may be no substitute for experience and proficiency with the software and tools used in the entire development process. It is one thing to discuss how a database should behave and even document these behaviors completely. Implementation and modification of these behaviors is an altogether different issue. The process of behavior implementation consists of the design and construction of a solution following the identification of the problem domain and the data model.

Because of the difficulties associated with the definition of a fixed set of current requirements and the determination of future utilization, the database problem domain is typically a moving target. The size and scope are constantly changing. In addition, insufficient identification of appropriate database 'behaviors', poor communication, and inexperience in the problem domain leads to inferior solutions. As a result, the solution domain rarely approaches an optimal solution. The database developer must attempt to develop a database model that closely matches the perceptions of the consumer, and deliver a design that can be implemented, maintained, and modified in a cost-effective way. A partial solution is more likely. The consumer will then dictate whether there is 1) enough of a solution to use, 2) the solution is of sufficient quality and, 3) whether they trust the database. Additionally, databases to be used in online analytical processing, data warehousing, or data mining applications present difficult challenges. The information consumer in these areas generally does not know what may be asked of the database. The database must behave in a fashion to respond to the most difficult requirement of all; that which the consumer has not yet thought of.

And finally, an additional important contributor to database quality that is difficult to categorize is that of information risk. Risk is addressed in the project management literature but not even discussed in the information quality literature. Risk may determine the grade of acceptable information quality. Consumers of on-line critical care database information that monitors hospital patients require a very high grade of information quality because the risk is very high. A database that tracks responses to a customer satisfaction survey, on the other hand, may be of lower grade because the overall information risk is low.

CONCLUSION AND RESEARCH DIRECTIONS

How does one ensure a final database product that is of high quality? Database quality must be measured in terms of a combination of dimensions including process and behavior quality, data quality, and model fidelity. The framework presented above offers a typology for assessing these dimensions. The purpose of this paper was to expand on the existing research on data and process quality in an attempt to provide a more comprehensive view of database quality. The area is of great concern as information becomes a critical organizational asset and preserving organizational memory remains a high priority (Saviano, 1997). Further research is required to validate the framework; to identify additional quality dimensions and develop metrics to quantify the properties; and to develop and deploy techniques to improve the fidelity of the data model.

REFERENCES:

Ballou, D. & Pazer, H. (1995). "Designing Information Systems to Optimize the Accuracy Timeliness Tradeoff". ***Information Systems Research***, 6 (1), 51-72.

Braithwaite, T. (1994). ***Information Service Excellence Through TQM, Building Partnerships for Business Process Reengineering and Continuous Improvement.*** ASQC Quality Press.

Chignell, M., & Parsaye, K. (1993). ***Intelligent Database Tools and Applications.*** California: Wiley.

Churchill, G. A. (1991). *Marketing Research: Methodological Foundations*, Dryden Press.

Costin, H. (1994). ***Total Quality Management.*** United States: Dryden.

Delen, G., & Rijsenbrij, D. (1992). "A Specification, Engineering and Measurement of Information Systems Quality". ***Journal of Systems Software***, 17(3), 205-217.

Dvir, R., & Evans, S. (1996). "A TQM Approach to the Improvement of Information Quality." *http://wem.mit.edu/tdqm/papers.html [1997, June].*

Dyer, M. (1992). ***The Cleanroom Approach to Quality Software Development.*** Wiley.

Fox, C., Levitin, A., & Redman, T. (1994). "The Notion of Data and its Quality Dimensions." ***Information Processing and Management***, 30 (1), 9-19.

Hoxmeier, J. (1997). "A Framework for Assessing Database Quality." ***Proceedings of the Workshop on Behavioral Models and Design Transformations: Issues and Opportunities in Conceptual Modeling, ACM Sixteenth International Conference on Conceptual Modeling***, Los Angeles, CA, November, 1997.

Hoxmeier, J. (1995). "Managing the Legacy Systems Reengineering Process: Lessons Learned and Prescriptive Advice." ***Proceedings of the Seventh Annual Software Technology Conference***, Ogden ALC/TISE, Salt Lake City, April, 1995.

Hoxmeier, J., & Monachi, D. (1996). "An Assessment of Database Quality: Design it Right or the Right Design?" ***Proceedings of the Association for Information Systems Annual Meeting***, Phoenix, AZ, August, 1996.

ISO, International Organization for Standardization. (1993). ***Quality-Vocabulary (Draft International Standard 8402).*** Geneva, Switzerland: ISO Press.

Levitin, A., & Redman, T. (1995). "Quality Dimensions of a Conceptual View." **Information Processing and Management**, 31(1).

Lindland, O., Sindre, G. & Solvberg, A. (1994). "Understanding Quality in Conceptual Modeling." **IEEE Software**, Vol. 11, No. 2, March 1994, 42-49.

Martin, J. (1976). **Principles of Database Management**. New Jersey: Prentice-Hall, Inc.

Menon, A., Jaworski, B. & Kohli, A. (1997). "Product Quality: Impact of Interdepartmental Interactions." ***Journal of the Academy of Marketing Science***, 25(3).

Moriarty, T. (1996). "Barriers to Data Quality." ***Database Programming and Design***, 61, May, 1996.

Navathe, S. (1997). "Conceptual Modeling in Biomedical Science." ***Proceedings of the ACM Entity Relationship 97 Modeling Preconference Symposium***, Los Angeles, CA.

Redman, T.C. (1995). "Improve Data Quality for Competitive Advantage." ***Sloan Management Review***, 36 (2), 99-107.

Rolph, P., & Bartram, P. (1994). ***The Information Agenda: Harnessing Relevant Information in a Changing Business Environment.*** London: Management Books 2000, 65-87.

Sankar, C., & Marshall, T. (1993). "Database Design Support: An Empirical Investigation of Perceptions and Performance." ***Journal of Database Management***, 4 (3), 4-14.

Saviano, J. (1997). "Are we there yet?" **CIO, 87-96**, June 1, 1997.

Schmauch, C. (1994). ***ISO-9000 for Software Developers***. ASQC Quality Press.

Standish Group. (1997). "The Chaos Report." *http://www.standishgroup.com/chaos.html.* [1997, November 7].

Storey, V., & Wang, R. (1994). "Modeling Quality Requirements in Conceptual Database Design." ***Total Data Quality Management, Working Paper Series:*** TDQM-02-94 *http://web.mit.edu/tdqm/www/wp94.html.* [1997, July].

Strong, D., Lee, Y., & Wang, R. (1997). "Data Quality in Context." ***Communications of the ACM***, 40(5), 103-110.