10

Managing with Less than Absolute Integrity

Amit Sheth
Large Scale Distributed Information Systems Lab
University of Georgia
http://lsdis.cs.uga.edu

Abstract of the Invited Talk

Our key position is that for less then perfect real-world information systems the standard notions of data correctness or integrity used in research literature are inappropriate and inadequate. Hence we call upon the researchers and practitioners to pay more attention to understanding less than absolute data integrity requirements of the users and the applications, and to developing techniques to support such requirements. We will discuss this position and the need to develop criteria and techniques to support "weaker" or application specific data integrity requirements with the help of following four situations facing information systems management:

- data quality in multidatabase systems
- consistency of related (interdependent) data in multidatabase systems
- data integrity in process-centric environments (utilizing workflow process automation), and
- data integrity involving heterogeneous digital media data, such as those increasingly found in Internet and intranets.

The important dimensions of data integrity that the above requirements address are:

- accuracy or correctness,
- completeness,
- consistency, and
- currentness.

This position paper gives the outline of my talk.

Keywords

data integrity, absolute integrity, real-world, environment, applications, databases

Integrity and Internal Control in Information Systems Volume 1
S. Jajodia, W. List, G. McGregor & L. Strous (Eds) © 1997 IFIP. Published by Chapman & Hall

1. INTRODUCTION

A significant amount of the database literature to date has focused on correctness and consistency of centralized or distributed databases involving structured data. We will discuss some real-world driven requirements for data integrity, in the context of data integrity (specifically correctness or consistency of data) in four key situations that arise in information systems management in the real world industrial environments. The first two are quality of data and consistency of data in multidatabase systems mainly involving structured data. The third issue is going beyond the database-centric views of information system (IS) management in 1980s, and discuss the challenges in managing data integrity in the context of process automation. Finally, we discuss the new problems faced in managing data integrity in the context of heterogeneous digital media data, access to which is increasingly enabled by Internet and Web related technologies.

We respond to the critical question of why traditional solutions to management of data integrity often fall short in meeting these requirements, as follows. In many real-world information systems, very general data integrity solutions cannot be devised, and we need to focus on meeting application specific requirements. Furthermore, we believe that there is much more to gain by understanding what level of data integrity is necessary, and how can we support adequate data integrity needs that are less stringent than the traditional criteria for data integrity in our imperfect world.

In Section 2, we discuss a couple of characteristics of the real-world computing environments that critically limit the applicability of the majority of research on data integrity. In section 3, we briefly discuss three of the four issues we have mentioned above. This discussion is based on [SRK92, SWK94] , which may be referred to for significant additional details including relevant literature.

2. SOME CHALLENGES OF THE COMPLEX REAL-WORLD TO THE IS MANAGEMENT

Static Databases versus Ever Changing Applications and Environments

A vast majority of database research has implicitly assumed static databases (that is databases whose definitions, descriptions or schemas do not change). Additionally, in practice, most databases were defined to serve a single application or a set of applications identified to be the user of the database prior to designing the databases (leading to the well known stove-pipe phenomena). However, in reality databases need to evolve for numerous reasons. Some of the recent work on schema evolution has addressed issues related to structural components, but the issue of how such an evolution affects integrity constraints associated with the schemas, remain largely unexplored. Another related issue is that of new applications that

are developed to utilize the databases that already exist. However, while a new application can use or share the databases as structured currently, many additional complexities may be introduced because of subtle semantic differences in interpretation and use of stored data, or due to additional, fewer or incompatible semantic integrity constraints imposed by the new application. One relatively simple consequence of sharing of data by multiple applications can be translated into the well- known view update problem. For example, if one application uses 4 out of 5 fields of an address table, the missing attribute value would pose a problem when this application updates the relation. Given that most view update algorithms have limited their attention to the Select-Project-Join view, database application developers and administrators are on there own when dealing with more complex relationships. The view update problem is probably among the better studied problems. There are hardly any methodologies, tools and techniques that address the problems of database and application evolution in the context of distributed, heterogeneous, and autonomous databases and distributed, multiple applications utilizing different databases. Two such issues are discussed briefly in Section 3.

Database (is not) at the Center of Universe

Much of the database research literature assumes the database management system (DBMS), or multiple DBMSs, to be solely responsibility for managing the integrity of data (for example, consider the literature on multidatabases and federated databases). However, in a vast majority of real-world computing environments, a DBMS is just one of many tools or system components. Of particular relevance to data integrity issues are various system components that support persistence and/or transactions. These components include persistent queues and persistent data stores (including those supported by communications infrastructure such as persistence service of CORBA, or associated with programming languages such as persistence supported by some Java implementations), and systems that support transactions such as Object Transaction Service of CORBA or Transaction Processing monitors, either as an independent component or increasingly as integrated with a Web server. Two observations are pertinent here:

- Many applications involve the use of multiple databases or information resources, and the corresponding distributed information processing involving participation of multiple system components of the types mentioned above.
- In 1990s, the IS management is increasingly taking an operation-centric or process-centric view as opposed to data-centric view that was prevalent in 1980s. Increasingly popular workflow management for business process automation support an application-oriented form of IS integration as opposed to data-oriented integration afforded by federated or multidatabase systems. For supporting data integrity, both so-called application-centric and data-centric components of information processing need to symbiotically participate. In this context, the concepts and techniques that evolve from data-centric perspective, such as the advanced transaction models, do not sufficient support the data integrity needs of ISs [WS97].

Absolute Integrity - Is it possible? Is it needed?

Much of the attention of database researchers involved with data integrity has focused on defining what we will term *absolute integrity*—that is the issue of data correctness and consistency as presented in terms of yes or no, with no middle ground. This involves various notions of consistency of data and use of the serializability criteria for centralized and distributed database transactions, and one-copy serializability criteria for replicated data. The issue here has centered around transforming databases from one correct state into another. Although these concepts and the corresponding techniques remain important within the context of a single database or several tightly-coupled databases, we feel that these are incapable of capturing the true complexity of IS management in the real world. More attention needs to paid to developing the solutions that reflect different degrees of gray rather than pure black and white, by answering the questions 1) when is data acceptably consistent or correct, and 1) how inconsistent is the data. We will discuss some examples along this line, but much more remains to be done.

Beyond Structured Databases

The vast majority of new data and applications in the increasingly Web-centric IS environments involve unstructured or new media (audio, image, video) data that are usually not managed by traditional databases. Semantically related information, for example about an entity such as a person, customer or a product, may be stored in various digital media in various independently managed resources. Early work on representing semantic correlation among heterogeneous media data has been proposed. One example is the proposed metadata reference link <MREF> as a logical complement to existing physical-level links in <A HREF>) and corresponding modular extensions to the Web-based infrastructure [SK96]. The hard questions related to the degree of consistency needed among these related data managed by heterogeneous and independent information resources, and the techniques for enforcing such consistency requirements, remain to be answered.

Metadata integrity itself is an important aspect of modern information systems. Metadata is representative of the information artifacts themselves with which integrity needs to be maintained. In many information systems metadata is the link between the users and the information and in such cases metadata integrity becomes more important. An interesting example would be the Web indexes where the full text index is the metadata. In most indexes we notice that maintaining consistency with the real world is becoming a non-trivial problem as the size of Web content continues to explode. Absolute integrity would be impossible to expect in such an environment and issues that need to be considered are how frequently to synchronize the metadata with the real world and to device methods for determining what are the relevance limits for integrity and then detecting these conditions.

3. TWO CASE STUDIES

We now use two cases in IS (specifically database) management to discuss the need for weak, application-specific forms of data integrity requirements.

Data Quality

Before we can apply any solutions for keeping data consistent, the current data needs to be of good quality. Unfortunately, that is not the case in most operational environments, as we learned through our study of and experiences with several client companies of Bellcore [KS93]. Examples of poor data quality include 1)errors in input data (e.g., a partial or nonexistent address), 2)data inconsistencies (e.g., different customer billing addresses for the same customer or incorrect Zip code for the location), and 3)unintended redundancy (e.g., multiple customer records because of different representations of the same customer such as DEC, Digital Equip. Corp., and Digital Equipment Corporation). These are often contributed to by duplicate data produced by different processes and organizations. Poor data quality is a result of a variety of factors, including flawed data acquisition and data creation processes, flawed data update processes, inability to enforce constraints among related data in multiple databases, duplicate data produced by different methods, organizations and processes, process re-engineering, and company reorganizations.

There are two aspects of addressing data quality management: data validation and data clean-up. *Data validation* refers to identification of data quality problems, for example, by identifying inconsistent or incomplete data in inputs from users or in the existing databases. *Data cleanup* (or purification) is the process of improving data quality (usually after the data validation identifies poor quality data) for example, by removing inconsistent data or making data more complete.

Not all aspects of data quality can be determined or enforced by computerized systems. However, any technology that can assist in addressing this problem needs to support:

- capturing business rules, practices and constraints that define data validation and cleanup rules, and
- integrating those rules with access to databases where a significant portion of corporate data reside.

Examples based on real applications, and one approach to addressing data quality problems using deductive database technology, are discussed in [SWK94]. While some clean-up operations can be performed automatically in a batch mode of operation, others may involve interactive human participation.

Management of Interdependent Data or Multidatabase Consistency

Many large companies use multiple databases to serve the needs of various application systems. One of the significant problems in managing these databases is maintaining the consistency of inter-related data in an environment consisting of multiple semi-autonomous and heterogeneous systems. We use the term *interdependent data* to imply that two or more data items stored in different databases are related through an integrity constraint that specifies the data dependency and the consistency requirements between these data items. Management of such data implies that a certain degree of mutual consistency among the interdependent data is maintained. Therefore, the manipulation (including concurrent updates) of the interdependent data must be controlled.

In the majority of existing applications, the mutual consistency requirements among multiple databases are either ignored, or the consistency of data is maintained by the application programs that perform related updates to all relevant databases. This can be accomplished using various techniques. For example, a message may be sent to another database system managing related data, so that a complementary transaction will be submitted there, or a replica of the data is sent to another system, either electronically or physically (e.g., a tape). However, these approaches have several disadvantages. First, they rely on the application programmer to enforce integrity constraints and to maintain mutual consistency of data, which is not acceptable if the programmer has incomplete knowledge of constraints to be enforced. Secondly, a modification of a part of an application requires changing of other parts of the same or another application to maintain integrity and consistency. Since integrity requirements are specified within an application, they are not written in a declarative way. If we need to identify these requirements, we must extract them from the code, which is a tedious and error prone task.

Alternative approaches to the problem are based on system supported maintenance of mutual consistency. A possible solution is to enhance the techniques of preserving integrity that were proposed for distributed databases. The main limitation of these techniques is that they assume that the consistency between the related data must be restored immediately. Such an *immediate consistency* criterion requires that as soon as a transaction completes, all related data are also mutual consistent. In the case of replicated data, one-copy serializability criterion has been usually used. However, in loosely-coupled environments we may need to temporarily tolerate inconsistencies among related data.

Several weaker consistency criteria that appear in literature include mutual consistency requirements using timing constraints, and the specification of coherency condition used to define "how far" primary copy and quasi-copies can diverge based on time, versions, and arithmetic conditions (please see [SRK92 for details). A mutual consistency criterion called *eventual consistency* states that related database objects are made consistent at certain points of time specified by a

condition, although they may not be consistent in the interim intervals. The condition is specified as a combination of time and data state (including events/operation). Eventual consistency allows the related objects to diverge during some period, as long as they will be made consistent periodically. Lagging consistency assumes that the data in one database may be most current while in the other ones the data may not be up-to-date. Updates applied to the first database are always propagated to the related databases. Hence, if all external updates are stopped, the databases will become consistent. Eventual consistency does imply that at some point in time, all databases will be consistent, while lagging consistency does not imply this, because some databases may always lag behind others. Some of the early approaches on enforcement of weaker consistency criteria appear in [KRS93, GKG97, SK97]. These, however, remain quite limited in the context of supporting real-world applications and IS environments.

4. PARTIAL BIBLIOGRAPHY

Among the publications that have begun to address some of the issues outlined in this position paper, the following is a small subset. Citations in the following publications can however lead to other relevant work.

[GKG97] D. Georgakopoulos, G. Karabatis and S. Gantimahapatruni, "Specification and Management of Interdependent Data in Operational Systems and Data Warehouses," Distributed and Parallel Databases, 5 (2), April 1997.

[KRS93] G. Karabatis, M. Rusinkiewicz and A. Sheth, "Correctness and Enforcement of Multidatabase Interdependencies" in Lecture Notes in Computer Sciences #759: Advanced Database Systems, N. Adam and B. Bhargava, Eds., Springer-Verlag, 1993.

[KS93] G. Karabatis and A. Sheth, "Specifying Interdependent Data: A Case Study at Bellcore", the Proceedings of the SIGMOD, Washington DC, June 1993.

[SRK92] A. Sheth, M. Rusinkiewicz, and G. Karabatis, " Using Polytransactions to Manage Interdependent Data" in Transaction Models for Advanced Database Applications, A. Elmagarmid, Ed., Morgan-Kaufmann, 1992.

[SWK94] A. Sheth, C. Wood, and V. Kashyap, "Q-Data: using deductive database technology to improve data quality", in Applications of Deductive Databases, R. Ramakrishnan, Ed., Kluwer Academic Press, 1994.

[SK96] A. Sheth and V. Kashyap, "Media-independent Correlation of Information: What? How?" Proceedings of the First IEEE Metadata Conference, April 1996.

[SK97] L. Seligman and L. Kerschberg, "A Mediator for Approximating Consistency: supporting "Good Enough" Materialized Views," Journal of Intelligent Information Systems, Vol. 8, 1997.

[WS97] D. Worah and A. Sheth, "Transactions in Transactional Workflows in Advanced Transaction Models and Architectures, "S. Jajodia and L. Kerschberg, Eds., Kluwer Academic Publishers, 1997