enterprise integrity

By DAVID McGOVERAN

Understanding Messages

Readers may have noticed my tendency to express concerns, an occasional new idea, and even opinions. Last issue's column included a few observations that require a bit more discussion. Perhaps most radical was my statement that labels for data elements do not solve the problem of semantics — however much I applaud the use of eXtensible Markup Language (XML).

Many of my colleagues involved in XML-worship disagree with me. Consider, however, a simple example of an XML tag, "Amount." Amount of what? Well, if the structure is nested, we might know this is a credit authorization. There might even be a tag: "Credit Authorization." But is this semantics? Do these tags provide meaning? Is this the amount requested, authorized, or a limit? Are we told how the data was derived? How it may change or be used in the future? No, we're not. Tags are just sequences of characters at best — with some context-free meaning in a natural language.

Even this "best" of possibilities is a bit strained. There are many natural languages, and natural languages are certainly not context-free. So how do we make the tags context-free? We do so by agreement (a.k.a. standards of interpretation). Unfortunately, developing such standards is difficult even within a single company, let alone throughout a business sphere of influence. The potential scope of standards development and enforcement is generally quite small (consider RosettaNet and BizTalk). Even then, difficulties remain with XML data interpretation, especially when it's used for business analysis across companies.

Contrary to what some believe, context and semantics are inextricably intertwined. If we specify the permissible usage of a data element, we specify the permissible contexts for it. This is the essence of specifying integrity constraints among data elements. Don't even suggest that integration brokers solve the semantics problem. Integrity constraints have little or nothing to do with data representation. Integrity constraints define relationships among complex abstract data types, permissible operations on those types, and the values that typed variables can have.

You may be familiar with integrity constraints from Structured Query Language (SQL) databases, but to publish these weak implementations, along with the XML schema, would be a marginal semantic improvement. SQL Database Management System (DBMS) products failed to support all the types of integrity constraints required by the relational model. (No, domain constraints still aren't supported.) Furthermore, relational semantics include proper collections of relations (i.e., properly designed tables). Relations exhibit implicit relationships known as dependencies. If we know all dependencies and constraints of a complete relational schema (model of a particular enterprise or application), we can determine the logical design of the database — and also the semantics.

Of course, there are other aspects of semantics that might not be fully represented in this way. Transaction properties — including permissible time to completion, the level of transaction isolation, and even the transaction model used — are also important. All these are specific to the use of the data involved and determined by semantics. For example, the required level of isolation cannot be determined solely by the request to execute a particular transaction. The design of the target database or databases, the mix of concurrent transactions, and the entire source and target business context must be included. Only then can we determine the conditions for "safe" transaction execution and whether or not failures are recoverable, given the mechanisms available. Were all this predictable, such semantics could be "hard-wired" into the application. But that wouldn't be much of an advance for the flexible world of e-business, would it?

Data and transaction integrity constraints define business semantics. This adds up to an open challenge to the Worldwide Web Consortium (W3C). We need a standard by which XML tags can be associated with complex models of (transactional) semantics. While businesses often consider data and transaction model details private, we still need the possibility of publication internally and with trading partners. Constraints, and the varieties of transactions, are too diverse for application-specific definition by some committee as "standard" business exchange transactions. A more general approach is needed — one that will permit each transaction participant to protect their data, while using that of others, in the most correct manner.

An XML constraints standard will permit self-defining semantics in addition to XML's laudable, self-defining formats. Truly automated data transformation for application-to-application (A2A) and business-to-business (B2B) integration would become possible — without database contamination, incorrectly run transactions, and misinterpreted messages. Until we can truly understand message meanings, enterprise integrity is always in jeopardy.

David McGoveran is president of Alternative Technologies, Inc. He has more than 20 years' experience with mission-critical applications and has authored numerous technical articles on application integration. e-Mail: mcgoveran@alternativetech.com.