

The PDFs of *eAI Journal's* Data Integration department are sponsored by:



Data Junction sets the standard for data integration with award-winning technology, customer support and technical services. Right now, more than 40,000 customers invested in Data Junction technology to solve their most pressing integration concerns, making our software the most widely deployed in the world. What integration challenges are you facing? Discover how Data Junction's solutions fit into your integration world. For an on-line demonstration of the power and versatility of Data Junction, contact us at 800.580.4411 or info@datajunction.com

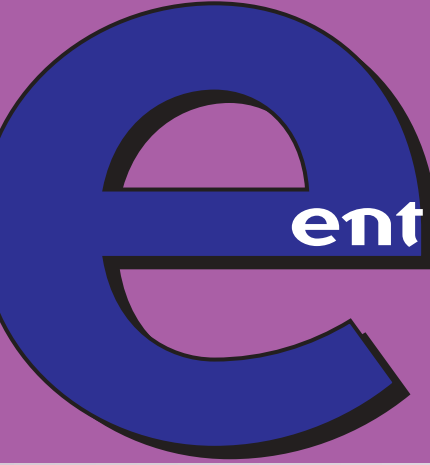


Data Junction Corporation
5555 North Lamar Blvd.
Ste. J-125
Austin, TX 78751
tel: 512-452-6105 ext. 256
fax: 512-467-1331
www.datajunction.com



This article is a PDF version of the one that appeared in a recent issue of *eAI Journal*, the leading resource for e-business, application integration, and Web services.

Integrating the Interconnected World™



enterprise integrity



By DAVID MCGOVERAN

Data Integration, Part VIII

Lowering the costs of integration and improving the incremental Return on Investment (ROI) depends on understanding and preserving data semantics. Using a greatly simplified theory of data type relationships, we've recently examined the ways in which understanding semantic relationships between source data types and target data type requirements establishes the semantic transformations required. Structural or syntactic transformation requirements are trivial. This month, we'll quickly consider two special, degenerate cases of semantic relationship and then wrap the series up with an incremental approach to data integration.

First, if the source data value is a member or instance of the target data type, we can infer that the source and target data types are equivalent and no semantic transformation is required. Second, if the source data type is semantically disjointed from the target data type, no semantic transformation is possible. Although this should be obvious, I have seen integrators — oblivious of semantics — syntactically transform a data source semantically inappropriate to the target application. Sometimes, the resulting errors are immediately obvious; at other times, they're subtle and more difficult to discover.

The primary types of semantic transformation are:

- Combining two or more fields having different data types
- Decomposing a field into two or more new fields
- Aggregating multiple values
- Disaggregating aggregate values
- Consolidation
- Synchronization
- Generalization
- Sub-typing.

As we've seen, these often overlap and multiple transformations may be required. We've simplified the issue considerably by considering relationships at a macro level, ignoring the fact that the semantic components of properties, operations, and constraints may distinguish differing relationships. Nonetheless, this analysis should help guide a more detailed effort, should you decide to pursue one.


As presented in the first few columns of this series, discovering semantic relationships "as needed" is a high-cost, high-risk, inefficient approach. On the other hand, developing an enterprise data model is a long-term, costly effort, almost always obsolete before it's deployed. So what can we do about this? The solution requires embracing the weaknesses in, and combining the strengths of, these two dominant approaches. The result is a Total Data Quality Management (TDQM) approach to an enterprise data model.

TDQM insists that data quality must be built-in, that it's the responsibility of everyone, and that it's the subject of continuous

improvement (not a "reengineering" or "big-bang" approach). TDQM must be adopted as a pervasive, corporate philosophy. It's guided, not owned, by the IT department. The essential step of TDQM is creating a dynamic metadata repository that acts as a container for data semantics as they're discovered and as a driving reference during integration projects. Here are the key principles of TDQM:

- Fill the repository incrementally, never "upfront." This is a pragmatic, not academic, effort.
- Ensure that the repository supports a theory of semantic types. It should define data semantics by capturing constraints and not just data syntax, and relate them to existing types through dependencies.
- Don't accept application software unless data semantics has been defined in an importable data model.
- Use versioning, partitioning, and type relationships to organize metadata. Never delete it.
- Commit to driving application development and integration projects from the repository.
- Use data integration tasks as opportunities to use, refine and validate the repository.

These principles let organizations develop and continuously improve a metadata repository with focus on near-term goals and without massive investment before use. Moreover, it provides objectively verifiable, reusable data semantics that significantly reduce data integration related efforts, costs, times to delivery, and maintenance. By preventing data semantic errors during data integration, the high cost of error recovery is avoided. All this translates to higher data quality and higher return on integration.

Still not sure TDQM is important? Consider the business implications if proper attention is not given to these semantic problems. In the case of counting customers mentioned in an earlier column, the CEO and the investors wanted to know how the company was doing in market share compared to its competition. This information would be used to decide whether or not to make an acquisition, expand existing market, or focus on customer retention. It would also support cost projections for maintenance, marketing, and customer support. Without accurate data in near-real time, none of these decisions could be made without tremendous risk. Bottom line? If you don't understand data semantics, you can't maintain enterprise integrity. 

David McGovern 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: mcgovern@alternativetech.com; Website: www.alternativetech.com.