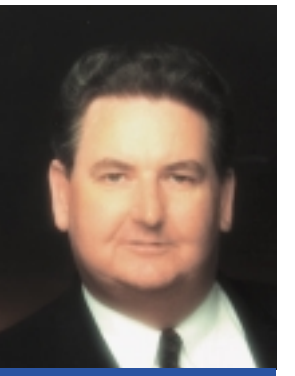
eAI JOURNAL
www.eaijournal.com

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.

enterprise integrity



By DAVID McGOVERAN

Data Integration, Part VI

Understanding the varieties and complexities of semantic transformations is essential to successful data integration. Previous columns have mentioned some situations in which semantic transformations arise. This column and the next two will review them in more detail. First, we need to clarify what we mean by the semantic aspects of metadata.

For any two data types, the possible semantic relationships are virtually unlimited. However, with a little simplification, we can focus on the important relationships. Each data type can be understood as being defined by a set of properties and operations that, together with a set of constraints on them, determine its permissible uses. Some of these properties may have a metric character, leading to measures and quantitative values associated with the data type. Other properties may have only a qualifying character, denoted but not quantified.

Two data types are semantically equivalent if they have the same defining properties and operations and the same set of constraints on those properties and operations. By contrast, syntactic equivalence only guarantees that two data types each have a representation with isomorphic structures, units, and values. For example, revenue and profit are syntactically equivalent but not semantically equivalent. Establishing semantic equivalence is the primary and most difficult task of data integration (syntactic equivalence being relatively easy).

Combining two or more fields having different data types implicitly involves a semantic transformation.

If the data type of a value produced by an application (the source) is not semantically equivalent to the data type of a value that's to be consumed by a second application (the target), the value must undergo a semantic transformation. The semantic transformation is determined by the semantic relationship between the data types of the produced and consumed values. Semantically, the source data type can be a subset, superset, member, disjoint with, or intersecting with the target data type with respect to sets of properties, operations, or constraints. A proper theory of type relationships would examine all these possibilities and their combinations. We'll simplify the matter by not distinguishing between property, operation, or constraint relationships. We'll only consider whether the source data type is a component, subtype, supertype, distinct type (disjoint with), has a subtype that's equivalent to a subtype, or has a supertype that's equivalent to a supertype of the target data type.

If the source data type is a component of the target data type, then the semantic transformation must consolidate the various components to create a proper instance of the target data type. The difficulty with consolidation is guaranteeing that relationships (constraints) among the components are respected. For example, if the target data type is an area, the component subtypes would be width and length. If the width and the length are obtained from two different sources, we must guarantee that they have the same context (refer to the same area) and that they are not both either lengths or widths.

Combining two or more fields having different data types implicitly involves a semantic transformation. This is true even in the simplest cases. For example, summing the quantities of items in a purchase order results in a data value of a new type — transforming a collection of item quantities into a quantity of items in the collection of items. If all these items are oranges, then we end up with the total quantify of oranges. This is an example of a special kind of consolidation called aggregation.

Another special kind of consolidation is time reconciliation, usually called synchronization. Data that has a temporal semantic component raises the possibility that the temporal context of the values might not be commensurate, let alone identical. This is often true for differential data, or data that captures change over time. For example, it makes no sense to combine quarterly revenues when some refer to calendar quarters and others to fiscal quarters. In practice, time reconciliation is difficult because time spans associated with differential data are seldom tightly controlled and even less often recorded with any precision. Even when these characteristics are overcome, there may not be sufficient information to transform the data so as to refer to a common time span. This part of the transformation requires separating aggregate values, then reaggregating them, a semantic transformation that's not uncommon even when other types of semantic relationships exist.

Next month, we'll examine the remaining dominant semantic relationships in our pursuit of enterprise integrity. eAI

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; Website: www.alternativetech.com.