OMICS A Journal of Integrative Biology
Volume 7, Number 1, 2003
© Mary Ann Liebert, Inc.

Applying Generic Schema Management to Bioinformatics

PHILIP A. BERNSTEIN

MANY OF THE DATA MANAGEMENT PROBLEMS faced by scientists involve the manipulation of database schemas and mappings between schemas. Example problems are as follows:

- Data translation—translating data from one format to another
- Schema evolution—coping with applications that have revised data format requirements
- Cataloging—offering raw and processed data in a format that other scientists can understand
- Schema integration—offering database access through a common schema that is mapped to heterogeneous databases
- Provenance analysis—describing all the transformations that have led to a given data product

For over 30 years, the database field has worked on tools to help with such problems. While much progress has been made, there are two major weaknesses that greatly limit the utility of this work.

First, most of the tools are oriented to just one of these problems and are useful only for one data model (e.g., SQL schemas, object-oriented schemas, or XML schemas). More often than not, database users who face such problems discover important differences between the exact problem definition or the data model required by their application and the capabilities of off-the-shelf solutions. As a result, they often end up developing customized tools. The database field needs to do better in providing robust and generic solutions that avoid the need for such customized approaches. If schema management tools were more generic, in the sense of solving a broad range of problems using a broad range of data models, they would be beneficial to a wider range of users. This would enlarge the market for such tools, which in turn would allow tool vendors to invest more in their development.

Second, current solutions require a lot of detailed database design activity, such as the development of field-by-field mappings, for which there is no automated assistance. Moreover, they require a lot of laborious object-at-a-time programming over complex object structures that represent schemas. These two facts account, in part, for users' willingness to forgo the use of off-the-shelf tools. That is, the tools often do not offer enough benefit to be worth living with their other limitations.

One approach to circumventing these problems is to identify generic schema management operators that treat schemas and mappings between schemas as bulk objects. Such operators could include the following:

- Match, which returns a mapping between two given schemas
- Diff, which returns the elements of one schema that do not map to another schema
- Merge, which returns an integration of two schemas based on a mapping
- Compose, which returns the composition of two mappings

For example, Match can help data translation and schema evolution by generating an initial mapping that a database designer can review and edit, which can then drive the translation or evolution. Merge can help cataloging and data integration by providing a global schema that subsumes schemas of several data sources. Compose can help offer data provenance by tracing through several mappings to show the data sources that were used to produce a particular data product.

Microsoft Research, Redmond, Washington.

Record-oriented data models have many obvious similarities. For example, they all break up a record into fields, possibly hierarchically structured. They may support different kinds of aggregation relationships, is-a relationships, an extensible instance-of hierarchy, and some built-in constraints, such as bounds on the cardinality of collection-valued attributes. By implementing the above operators on a suitably rich data model that has most of these features, the implementation can potentially be applied to most popular data models with little or no customization.

These operators, and others at a similar level of abstraction, can help address the main weaknesses of current tools that were noted earlier: They address genericity by making it easier to build schema management tools that address a broader range of application problems on a broader range of data models. The Match operator addresses the problem of database design effort by providing automated assistance, thereby saving the designer's time and potentially reducing the skills required to develop the mapping. And the operators address the problem of laborious object-at-a-time programming by manipulating schemas and mappings as bulk objects. A proposal along these lines, called Model Management, can be found in Bernstein (2003) and at the project web site (Model Management, 2003).

There is active database research to improve the semantics and implementation of the above operators and to develop new ones. Much of the work is focused on commercial applications, which is a good thing for scientists in that it provides another high-value market that will encourage vendors to pursue the development of these operators. Some work is also focused on the bioinformatics area. For example, we are currently applying the match operator to the problem of aligning two large ontologies of human anatomy. To exploit and accelerate this development, it would be valuable to identify application problems in the area of molecular and cell biology that these operators can address.

REFERENCES

BERNSTEIN, P.A. (2003). Applying model management to classical meta data problems. In: *Proceedings of the Conference on Innovative Database Research.* pp. 209–220. Available: www-db.cs.wisc.edu/cidr/program/.

MODEL MANAGEMENT. (2003). Project web site for Model Management [On-line]. Available: http://research.microsoft.com/db/ModelMgt/.

Address reprint requests to:
Dr. Philip A. Bernstein
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399

E-mail: philbe@microsoft.com