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

Will Database Systems Fail Bioinformatics, Too?

DAVID MAIER

THE DATABASE SYSTEMS RESEARCH COMMUNITY has always been eager to try to extend database technology to address application areas not well served by current commercial products: Computer-Aided Design, expert systems, workflow, financial analysis, imagery. Such efforts have almost always led to innovative developments in the past: object-oriented databases, deductive databases, active databases, sequence data models, array algebras. Yet DBMS penetration into the motivating areas has been modest at best.

THE TYPICAL CYCLE

Work on database technology for new application areas seems to follow the same basic cycle.

Identification

The limitations of existing DBMS products for a particular application area are articulated. Anecdotal evidence, conference panels, agenda-setting workshops elaborate the specific shortcomings of current technology, detail the challenges involved, and suggest productive research directions.

Investigation

The research community responds earnestly to these summons, and alternative data models, advanced access methods, new query languages, novel transaction services, and specialized evaluation techniques proliferate.

Implementation

Promising ideas get carried forward into working software. Sometimes that software is robust enough to support a particular application in the area of interest. Tests and benchmarks show the superiority of the new approaches. Some of the prototypes actually get commercialized, or their features are picked up in existing DBMS products.

Practice

Most applications in the area continue to use files. Perhaps a DBMS is used as a directory or catalog.

WHY?

Is there a reason that all the wonderful database technology developed in response to the needs of a particular application area isn't enthusiastically adopted? I speculate on several reasons:

Department of Computer Science and Engineering, Database and Object Technology Laboratory, Oregon Health & Science University, Beaverton, Oregon.

- Design theory and software development techniques don't exist for the new features. The researchers who created new capabilities in response to the needs of a particular application probably know how to apply them for that application. But it may not be obvious at all to developers working on other applications how to select and use these capabilities.
- Databases can't be integrated into existing tools. The end users may have tool suites that they currently use, but for which they do not possess or control the code base. Thus, they lack the means for connecting the existing tools to new data sources. Even when the code base is available, there may be other impediments, such as lack of an API for FORTRAN.
- The application area has changed significantly since the database research began. The functionality and performance needed may have increased greatly. Substring matching over a megabyte becomes approximate matching over a gigabyte, then morphs into statistical pattern recognition over a terabyte.
- Systems are hard to configure. Even when using basic features, database systems can be notoriously difficult to tune for best performance, especially when application characteristics are dynamic. Use of advanced features only aggravates the problem.
- Database technology delivers only a partial solution. A system that solves, say, three of five major issues for an application area is unattractive, leaving significant work for application developers. They may decide to custom code the whole thing, rather than deal with a hybrid solution.
- Markets are limited or fragmented. The commercial demand might not be large enough or uniform enough to justify bringing out a specialized product.

CAN WE DO BETTER THIS TIME?

Are the prospects for data management research becoming widely used any better for molecular and cell biology than for the dozens of other application areas we've tried to help in the past? Maybe not, but that won't prevent me from offering some suggestions.

Redefine the product

Perhaps we should emphasize producing people who understand both database technology and the application domain, rather than new software systems. People who are adept at building bioinformatics systems with existing technology are probably more helpful than new technologies that no one knows how to use. They are also likely to become early adopters of novel capabilities that become available. (By this measure, past database research efforts perhaps have been quite successful.)

Develop design techniques

We need to invest in learning how to systematically construct applications using the new concepts and implementations we develop. The process should include both logical and physical design, adaptation of legacy tools and data sources, application engineering and maintenance concerns.

Accept point solutions

Perhaps we shouldn't set our sights too high to begin with. It may be best to have a period of constructing solutions for individual researchers or groups before trying to produce generic technology. We may need many points in application space before the surface we are targeting becomes clear.

Hide the database

Maybe presenting data management capabilities at the usual level of a schema definition and a query interface is not useful. Conceivably, embedding the database in a programming environment that better matches the needs of bioinformatics computations and searches will be more productive. Here I am not so much thinking of a particular programming language as of a package such as S-PLUS™ or Mathematica™.

Understand hybrid systems

Possibly the best way to manage bioinformatics data is with a combination of files, DBMSs and other storage technologies. While overarching principles for hybrid architectures may be hard to come by, enough such systems exist that at least the common design patterns could be extracted and documented.

I don't know whether these specific suggestions will help us be more effective, but it behooves the database research community to think about the path into practice for what we develop.

Address reprint requests to:
Dr. David Maier
Department of Computer Science and Engineering
Database and Object Technology Laboratory
Oregon Health & Science University
20000 North West Walker Road
Beaverton, OR 97006-8921

E-mail: maier@cse.ogi.edu