ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Paolo Atzeni and Val Tannen (Eds)

Database Programming Languages (DBPL-5)

Proceedings of the Fifth International Workshop on Database
Programming Languages, Gubbio, Umbria, Italy, 6-8 September 1995

Paper:

Data Mapping and Matching: Languages for Scientific Datasets (Panel Report)

Paris Kanellakis

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

Data Mapping and Matching: Languages for Scientific Datasets (Panel at DBPL 95)

Paris Kanellakis *

Brown University

Fall 1995

Abstract

This is a report on a panel at the Fifth International Workshop on Database Programming Languages, September 6–8, 1995, Gubbio, Umbria, Italy. The panel was well attended and there was a fair amount of interaction with an audience of about 50 people. The panelists spoke for about 10 minutes each. Then, there was a 30 minute period involving questions and discussion with the audience. The order of the panel presentations was: Paris Kanellakis (panel chair) of Brown, David Maier of OGI, Peter Buneman of UPenn, Stan Zdonik of Brown, and Sophie Cluet of INRIA. We present the general theme of the panel, summaries of the panelist remarks, and a summary of the general discussion.

Introduction

The panel chair introduced the panelists and presented the panel theme. He set the stage for the discussion by contrasting commercial relational database technology with scientific data management. Database management systems have been very successful at providing efficient access to large databases of business applications. This success has been achieved for highly-structured record-oriented data by combining an elegant formalism (logic-based languages and algebras) with efficient implementation. New data-intensive applications such as those of the scientific community require efficient access to massive amounts of data, which differ in their semantics and organization from business data: (i) the data structure is more complex (complex objects, extensibility, heterogeneity, a fair amount of metadata); (ii) the time/space dimensions are essential (although poorly captured by the relational model); (iii) the querying of data often involves data mining for similarities.

Panel Topic: A primary motivation for new database technology is to facilitate classification and exploratory search of the broad spectrum of multimedia data, available both at a user's site and through network access. Many of the available datasets are scientific, residing in conventional databases or in, the more common and general, data exchange (DX) formats. The impact of current database technology (both object-oriented and relational) on managing scientific datasets is limited by a lack of interoperation with the growing variety of heterogeneous DX formats. Another significant problem of current database systems is insufficient modeling support for metadata as well as for spatial and temporal features, which are present in the majority of scientific applications.

This panel will discuss these limitations of existing database languages and explore fresh approaches towards information integration (data mapping) and manipulation (data matching).

Declarative Languages: From Relations to Constraints

Paris Kanellakis also briefly commented on the evolution of declarative data models from relational to constraint-based. Constraint databases are a candidate formalism for expressing, in a declarative fashion, queries on spatial and temporal

*Paris Kanellakis, his wife and their two children died unexpectedly and tragically on December 20, 1995. This is a tremendous loss to the DBPL community. An obituary is included in the preface to these proceedings. P.A. & V.T.

data. The tuples of the relational model are generalized to conjunctions of constraints. The underlying principle is to use in database languages, data types that are closer to the natural language specification of many application.

The motivation for more general data models (such a constraint-based models) has been increased functionality. Similarity queries over time-series data is a good example of needed increased functionality. The queries that one would like to express here involve detecting similar sequence patterns. For example, an exact match between two sequences is rare, but sequences may almost match or they may present *similarities*. The detection of these similarities has many applications from financial (e.g., stock prices) to earth science (e.g., temperature readings).

Data Exchange Formats and 3-Level Architecture

David Maier considered *data exchange* (DX) *formats*. The majority of scientific datasets do not reside in conventional databases but rather in DX formats (e.g., CDF, HDF, CIF, FITS, ASN.1, Express, etc.). These self describing data were developed for allowing programs to exchange data. This is possibly the fastest-growing form of network accessible data.

Some DX formats are now also being used for logical data definition and as the primary form of data storage. They are usually equipped with application program interfaces (API's). And, not surprisingly, data management functionalities such as access methods (e.g., records in netCDF), catalog, query facilities, are been added to these API's.

DX formats present a number of advantages, that are making them popular, the most crucial probably being the existence of link libraries specific to scientific domains. Indeed, DX formats are becoming standard in scientific communities. However, they are missing many features commonly found in database management systems. In particular, they do not scale up and their query facilities are very primitive.

David Maier advocated the use of an object-oriented database system as a *Hybrid Data Manager*. This is some middleware between the applications and the data sources (databases or files). This leads to a 3-level architecture with heterogeneous external sources at the bottom level, the object database acting as a mediator, and a homogeneous domain schema at the higher level. David Maier reported some experiments in Materials Science with the Gemstone object-oriented databases management system and 5 sources (two databases and 3 DX formats).

Genome Databases and Database Languages

Peter Buneman considered the case of *genome data* (e.g., ASN.1), an excellent demonstration of the adoption of DX formats despite their drawbacks. As in other fields, there are many reasons for this: (i) genome data is not adequately modeled using traditional database models, (ii) data descriptions/schemas are enormous and very rapidly changing, (iii) interoperability with special purpose algorithms (e.g., Blast or Fasta) is crucial. So, when developing the first genome banks, long-range concerns such as transaction-oriented support offered by database systems, were often overshadowed by economic as well as scientific pressure to get sequencing information in electronic form very fast.

To answer the needs of genome databases, database systems have to offer better linguistic support of collection type and other types (e.g., variants) that are encountered in DX formats. It is important to be able to ask complex queries spanning multiple databases. (E.g., find the information on the DNA sequence known to be Chromosome 22 between location 22p11.2 and q12.1; and for each sequence, identify similar sequences from other organism.)

OODBMS Support

Stan Zdonik questioned whether *object-oriented databases management systems* were capable of supporting the new scientific database applications of the 90's and considered a number of issues that this raises.

First, in OODBMS, the schema is considered to be very stable. This is not surprising in the view of the standard database applications: A schema is first designed; data is loaded or created; the schema does not then change much. In many scientific applications, the separation between schema and data is less clear. Indeed, in some cases, data arrives first (e.g., maps) and schema information are derived from the data. In general, much more flexibility is expected in terms of schema definition and also in the way real data is mapped to virtual data (view mechanism).

Queries in the scientific setting are the basis of data exchange. We have to deal with issues such as translation (data may be residing in files), multimedia types, interoperability with scientific libraries. The notion of query has to be enriched to be able to name resources (URL's), specify dynamic links, query complex bulk types (eg., patterns exploiting the structure of the collection), approximate queries.

In conclusion, Stan Zdonik stressed the particularities of query optimization needed for these scientific applications. A particular problem is that resources are distributed and there is no global system catalog. As a result, query optimization has to be interspersed with query execution.

Query Language and Middleware

Sophie Cluet also considered the support of DX formats by OODBMS. She treated more specifically the query language issue and the need for a middleware-based approach. She used as a motivating example a system developed at INRIA (with Serge Abiteboul, Tova Milo and others) for managing structured text.

DX formats may be queried using declarative query languages in the style of the standard for OODB, the *object query language* OQL. However, the OQL data model has to be enhanced, e.g., to allow for heterogeneous collections (variants). The query language itself should incorporate features such as access by content (information retrieval style) or by browsing. It is also important to be able to query data without complete knowledge of its structure.

Sophie Cluet also argued that one should abandon the hope to capture within a database system the needs of all possible applications (in particular, all multimedia applications). It is therefore essential to be able to interoperate with application programs. The various data sources have to exist with their own representations and the OODBMS in the role of a mediator/integrator provides a homogeneous view of the data. (This is in the style of the 3-Level Architecture of Dave Maier.)

This raises the issue of the choice of the data model for the mediator. This also implies that the same data will have possibly several representations (e.g., DX format at the source and object-oriented database in the database) materialized or simply virtual. This highlights the needs for mappings between these representations, translation of data/queries, propagation of updates; and of optimization techniques to support these multiple representations.

Discussion

The panel generated a very lively discussion:

Munindar Singh: What is the boundary between the database system (for scientific multimedia data) and application programming?

The point here is to delimit what functionalities are the responsibility of the system and what should be the user's responsibility.

The panel felt that the answer depended on the particular features (concerning multimedia data) supported by the database. The responsibility of the system is clearly extended to those features. For other (perhaps important) features, the responsibility is that of the user.

The panel also felt that extensibility of database operators was of key importance for extending the functionality of the database system.

Rakesh Agrawal: There is no indexing technology to index all features. Domain scientists have to understand the limitations of what is available. The IBM project Qubic was described as an example, where particular care was given to relating features extracted to the general pattern matching questions asked.

Jose Blakeley: Do all mapping/matching problems have the same flavor, or can they be classified in qualitatively different categories?

The example of version control of structured documents was contrasted to detecting a hurricane pattern in a satellite image.

The terms mapping and matching characterize two different activities, one a static organization of data and the other a more dynamic activity of detecting similarity patterns.

Guido Moerkotte: What are the database language core contributions to scientific data management?

The answers varied from interfaces, OQL queries, to extensible optimization.

Catriel Beeri: Is there a potentially very large number of ADT's in the area of scientific data management or not?

It was felt that about 10 bulk types could account for 99% of applications. If this was so, then there would be some hope that understanding how to optimize over these bulk types would make a big difference.

Data Mapping and Matching: Languages for Scientific Datasets (Panel at DBPL 95)

It was also pointed out that some important issues were not addressed by bulk type technology, e.g., similarity queries.

Limsoon Wong: Feedback and progressive refinement are standard techniques in information retrieval. How are these facilitated by database languages?

The panel felt that database programming languages should be designed so that such techniques are easy and natural to use through the language.

Val Tannen: Were pre-relational databases data exchange formats?

Some felt that there were common aspects, although pre-relational data models were much simpler than data exchange formats.

Subsequent discussion did not address the question directly. Instead it focused on whether a better understanding of data exchange formats and of the translation between them could take advantage of using “mediating” data models. The data model would serve as a reference to explain the various DX formats.

There was also discussion on the relationship of SGML with various data models.

The next question was posed by a member of the panel.

Dave Maier: Should a “mediating” data model support 1-d arrays and multi-d arrays?

This technical question illustrated a basic limitation of relational technology (with its emphasis on sets) to provide the mediating model between data exchange formats. Arrays would be critical when considering languages to specify transformations between DX formats.

If we think of the various features found in DX formats as integrity constraints. Then the choice of a “mediating” data model amounts to the selection of hard-coded integrity constraints. Now, it is not clear how much semantics should be included into the model. What would be left out would be left to the responsibility of the applications.

This brought the discussion back full circle to where the “mediating” data model would draw the line between database system support and application programming.