Multidatabase and Interoperable Systems

A. R. HURSON[1] and M. W. BRIGHT[2]

[1]*Department of Computer Science and Engineering, Pennsylvania State University, University Park,* PA 16802, *and* [2]*Administrative Information Systems, University of Pittsburgh, Pittsburgh,* PA 15260

Most large organizations have many different kinds of hardware and software. There are various large and small databases scattered throughout the network with different data models and access languages. Each user has a local data source or two that is strategic to their function. However, they also realize that there is more information out in the network, somewhere, that they need. These organizations need a data access system that can accommodate preexisting, heterogeneous, data sources scattered through the network. Multidatabase systems provide a uniform global database interface to exactly these types of systems. This global interface hides all the underlying distribution and heterogeneity of information in the system, so that users can submit a single query to capture data from any or all available databases. In addition, multidatabase systems allow participating local databases to remain autonomous; i.e., the local DBMS determines how to structure local data, what functions to expose to the multidatabase, what global queries it will process, and when it will join or leave the global system. This property of local autonomy means the multidatabase system provides global information access, but the organization does not have to replace its hardware, software, or user training. Organizations keep all their existing function, while adding significant global access capability.

Multidatabase systems have their roots in traditional distributed databases, but were significantly influenced by the rise of networks and the connectivity of heterogeneous systems. Traditional distributed databases are designed top-down—the global function controls local DBMSs (which are typically all the same system—homogeneous). Multidatabases are designed bottom-up from preexisting, heterogeneous DBMSs.

Multidatabase research started with Multibase in the late 1970s. Initial research interests focused on transformations between local data models and access languages to a different global data model and access language. Most of these systems used a global schema which represented a single, comprehensive description of all the local database schemas. Global query processing, including all the required transformations, was the main function available. In the mid-1980s, problems with creating and maintaining global schemas became evident. Multidatabase language systems (such as MRDSM) and federated databases (such as Heimbigner's system) suggested access language functions and partial global schemas (obtained through negotiation protocols) as solutions. Early systems were typically read only, because the problems of global concurrency control with autonomous, heterogeneous local DBMSs were very difficult. In the late 1980s, a variety of research began to address global concurrency control. Many approaches suggested new ideas of transaction management and explored new methods of defining correct concurrency based on the semantics of global applications. Up to this point, multidatabases had been predominantly described in terms of the relational data model. In the late 1980s, the object-oriented data model became the majority choice as the multidatabase global data model. Today, much research focuses on resolving the heterogeneous semantics of participating local databases. Another important aspect of current multidatabase projects is refining the earlier research into engineering problems/solutions so that the systems are robust enough for production use and commercial distribution. Some multidatabase prototypes are used as internal production systems. Some multidatabase

Integrated Computer-Aided Engineering, 2(1) 1–2 (1995)

CCC 1069-2509/95/010001-02

features are creeping into existing commercial DBMSs (such as Sybase and INGRES/Star). Mermaid is the first full multidatabase system to be commercially available and is marketed under the name InterViso.

This issue provides a glimpse into the state of the art in multidatabase research and engineering. We have a good understanding of how to syntactically transform database schemas between different data models. We do not have a good understanding of how to represent and transform the semantics of independently designed, heterogeneous database schemas. The first two papers, "A Fundamental Framework for Realizing Semantic Interoperability in a Multidatabase Environment" by Kiyoki, Kitagawa, and Hitomi and "Relational to Entity-Relationship Schema Translation Using Semantic and Inclusion Dependencies" by Ling and Lee*, address the problem of semantic heterogeneity. The object-oriented data model and the concept of data views are powerful tools for representing and integrating disparate databases. "An Extensible View System for Multidatabase Integration and Interoperation" by Yen and Miller* and "Schema Integration and Query Processing for Multiple Object Databases" by Chen, Koh, Kuo, and Liu present a system for integrating multidatabases systems using these tools. Large data repositories are often difficult to represent at the user interface because of the large amounts of related data items. "Structured Display and Browsing of Documentary Information" by Korfhage, Nuchprayoon, and Parmanto reviews several user interfaces which abstract large text databases so that they are amenable to user searches. Although concurrency control and query processing have a long history of research in the multidatabase field, there is always room to improve on the theory and mechanics. "Reliable Nested Transaction Processing for Multidatabase Systems" by Kang and Keefe and "Multiway Equijoin Query Acceleration Using Hit-Lists" by Scott, Perrizo, Gustafson, and Schilling address new methods of concurrency control and efficient query processing (respectively).

This issue is the result of a very difficult selection process. We received 29 excellent papers on various aspects of multidatabase engineering and 7 papers were selected for the special issue. We want to again thank the referees listed below who contributed greatly to the selection process and to improving the papers printed here. Quality publications would not be possible without the insight and patience of good referees. Finally, we would like to thank the Editor-in-Chief for his patience and guidance during the course of this project.

REFEREES

K. Barker	University of Manitoba
A. Bestavros	Boston University
D. Bradshaw	IBM
M. Bright	IBM
S. Brown	University of Tennessee
J. Carruthers	IBM
M. Carson	IBM
D. Chandler	IBM
Y. Chehadeh	Penn State University
K. Dash	Penn State University
B. Desai	Concordia University
M. Eich	Southern Methodist University
B. Eubanks	Hood College
B. Finch	IBM
O. Frieder	George Mason University
J. Gerrity	IBM
E. Gordy	IBM
P. Gowen	Hood College
A. Goyal	Virginia Commonwealth University
M. Hoferek	IBM
L. Hollaar	University of Utah
T. Horowitz	Hood College
T. Keefe	Penn State University
R. Korfhage	University of Pittsburgh
D. Lee	Ohio State University
G. Luckenbaugh	IBM
L. Miller	Iowa State University
I. Olasupo	IBM
S. Pakzad	Penn State University
M. Ranganathan	IBM
B. Shirazi	University of Texas at Arlington
C. Staton	George Mason University
M. Templeton	Data Integration Inc.
V. Ventore	MITRE Corp.
C.-H. Yen	Iowa State University

* Due to space limitations, this article will appear in Vol. 2, No. 2, 1995.